IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 November 2023, accepted 14 December 2023, date of publication 18 December 2023,
date of current version 27 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3344218

RESEARCH ARTICLE

Safe Autonomous Exploration and Adaptive Path Planning Strategy Using Signed Distance Field

HEYING WANG[1], **YUAN LIN**[1], **WEI ZHANG**[1], **WENTAO YE**[1], **MINGMING ZHANG**[2], **AND XUE DONG**[1]

[1]China-U.K. Low Carbon College, Shanghai Jiao Tong University, Shanghai 201100, China
[2]Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai 201100, China

Corresponding author: Xue Dong (xue.dong@sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52376160 and Grant 52006137, and in part by the Shanghai Sailing Program under Grant 19YF1423400.

ABSTRACT Autonomous exploration in unknown environment has remained challenging due to unexpected collisions, stuckness and slowdowns around obstacles. This paper reports a novel approach based on Signed Distance Field (SDF), to optimize path planning algorithms and autonomous exploration strategy for safe and adaptive navigation in search and rescue missions. A quantitative criterion is established for evaluating the safety of planned trajectories. Simulation results show that the proposed SDF-A* path planner outperforms traditional methods with a 30.10% increase in path safety (i.e. average distance from robot to obstacles) and a 64.11% reduction in time consumption; The proposed SDF-based Safe Autonomous Exploration Strategy, combined with SDF-A* path planner, outperform traditional methods, leading to significant increases (47.06%) in path safety and reductions (44.75% and 15.32%) in exploration time and path length, respectively. The viability, efficiency, and safety of the proposed methods are further validated through real-world experiments on a three-wheeled differential steering robot equipped with Jetson Nano and RPLIDAR-A3 lidar. Results show that the proposed approach adapts to different indoor environments and map configurations without prior parameter settings.

INDEX TERMS Path planning, autonomous exploration, path safety, signed distance field.

I. INTRODUCTION

Robots and autonomous moving vehicles have been increasingly used in search and rescue missions in post-disaster environments. These scenarios are often complex and with no priori map available. In order to facilitate efficient rescue operations in such challenging scenarios, it is essential for mobile robots to venture into unknown areas and autonomously explore them in a safe and efficient manner. By doing so, they can generate accurate indoor maps that can significantly assist rescue.

The autonomous exploration task can be divided into 4 subtasks: (a) mapping, (b) localization, (c) next goal assigning, and (d) path planning. The first two subtasks can be achieved using SLAM (Simultaneous Localization and Mapping) algorithm, while the latter two subtasks are usually realized by frontier detection [1] and classical path planning model like A* [2], respectively. Classical strategies employed in goal assignment and path planning typically prioritize exploration efficiency, with less consideration on robot safety. However, it is crucial to ensure safety of the robot while exploring unknown environments, especially in the context of search and rescue missions, which in turn also benefit exploration efficiency.

Traditionally, autonomous exploration methods can be broadly categorized into frontier-based methods and sampling-based methods. Frontier-based exploration strategies, frequently utilized in robotic exploration, automatically assign the next goal towards frontier edges [1], [3]. On the other hand, sampling-based methods have also been employed, aiming to sample the ''next-best-view'' for exploration. Frontier edges are defined as lines that distinguish known from unknown spaces in an occupancy grid map. Upon detection of a frontier edge, a point on the edge is

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

assigned as the next goal for autonomous exploration. Many previous research on optimizing the next goal assignment strategy focus on maximizing information gain or decreasing the exploration time [4], However, as the information gain does not account for the spatial distribution of obstacles, exploration goals near obstacles are often selected, resulting in collisions, low speed of movement, and even being stuck, all of which reduce the efficiency of exploration [5], [6]. Therefore, ensuring safety of next goal assignment and avoiding being trapped should have been the priority in an autonomous exploration task.

As a classical global path planner, A* generates the shortest path without considering the distance from the robot to the obstacle, therefore, in practical application, the robot can be too close to the obstacle [7]. To address the limitation of the A* algorithm, a common approach is to employ inflation layers in Costmap [8], which consists of a static layer, an obstacle layer, an inflation layer and a master layer. Inflation layers are commonly employed by the occupancy grid map in the Robot Operating System (ROS) navigation framework [9] to create a safety buffer around obstacles. As pointed out in [10], the computational load associated with the Costmap are primarily caused by the inflation layer. When updating the global Costmap, the updated area of the master layer in Costmap encompasses the entire two-dimensional grid map. Consequently, each update of the master layer inherently leads to the inflation of the entire two-dimensional grid map, significantly reducing computational efficiency, although inflation layers in Costmap can improve safety in robot path planning [11]. For example, when the robot approaches or enters the inflation layer, it starts to decelerate [8], reverse or even spin around, which is considered as inefficient behaviors, as is shown in Fig. 1. These inefficient behaviors, including unnecessary reversing, can increase the trajectory overlaps and time consumption, which not only diminish the efficiency of autonomous exploration but also increase the probability of the robot being trapped. This is because the inflation layer decelerates and restricts robot movement, even causing it to come to a halt. Consequently, avoiding reverse around obstacles becomes instrumental in reducing the probability of entrapment. As such, quantitative measures is also of great significance to evaluate the performance of the strategies on the above-mentioned aspects, in addition to some qualitative analyses that have been reported for autonomous exploration processes [12].

In recent years, Signed Distance Field (SDF) has been widely utilized in 3D reconstruction and path planning of Unmanned Aerial Vehicles (UAVs) [13], [14], [15], and has been shown to be a robust map representation for dense reconstruction of the environment and for trajectory planning. SDF is a mathematical representation of the surrounding environment that presents the distance to the closest obstacle at each point in space, with the sign indicating whether the point is inside or outside the obstacle. In 2D environment, Fossel et al. [16] have demonstrated that SDF provides higher accuracy in comparison to occupancy grid maps. However, limited research has reported on combining SDF information with 2D path planning and autonomous exploration.

In light of these backgrounds, the challenges and limitations in autonomous exploration and path planning can be summarized as follows: 1) inefficient behaviors during robot movement especially around inflation layer, 2) unreachable next goal assigned by autonomous exploration module, 3) lack of a quantitative criterion to evaluate safety of planned paths. These three factors are the focus of this study.

This work therefore aims to propose a safe and efficient next goal assignment and path planning strategy that incorporates SDF to replace the Costmap inflation layer. The performance of the proposed approach is evaluated in safety, efficiency (time consumption), coverage ratio (the ratio of explored area and the total area) and failure rates etc.

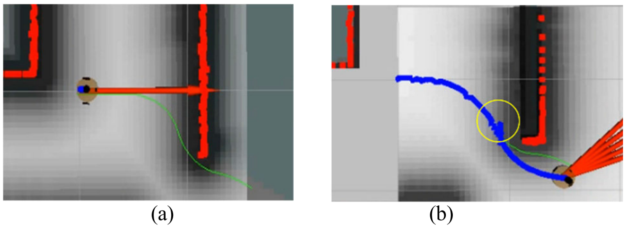


FIGURE 1. Demonstration of the limitations of classical A* planner. (a) Path planned by classical A* (green line). (b) Trajectory (blue line) with inefficient reverse (in yellow circle).

II. RELATED WORK

This section reviews the previous work in autonomous goal assignment strategy, path planning and path safety in autonomous exploration.

A. GOAL ASSIGNMENT STRATEGY

The frontier-based method [1] is widely applied in autonomous exploration due to its simplicity and ease of implementation. Keidar and Kaminka [17] present a Wavefront Frontier Detector (WFD), which employs two breadth-first search (BFS) approaches to identify frontier boundaries. However, as was discussed above, the frontier-based method can easily end up with unreachable target or leaving the robot trapped. To overcome this limitation, researchers have proposed algorithms to assign the next goal with higher level of reachability and safety. For instance, Senarathne et al. [18] propose an incremental approach called Safe and Reachable Frontier Detection (SRFD) to generate a valid frontier by inflating the obstacle which avoids a separate filter to remove invalid frontiers. Sun et al. [19] also aim to improve the reachability by detecting frontiers of the submaps and inflating them. However, similar to Costmap's inflation layer, this method removes the candidate goals that fall within the inflated areas, which may lead to information loss of map and low coverage ratio. At the same time, many research focuses on increasing the information gain or utility of goal assignment [5], [6] or decreasing the exploration time [4], however, relatively little attention is given to ensuring the safety of assigned goals. Furthermore, Zhao et al. [20] propose an approach that combines frontier-based and sampling-based exploration for Unmanned Aerial Vehicles (UAVs). In their method, the candidate next view is sampled from the map frontier. Based on the principles of frontier theory, exploration strategies based on rapidly-exploring randomized trees (RRT) unitizes the randomized tree expansion to detect and prioritize the frontiers [5], [21]. However, when it comes to search and rescue missions, the region to be explored is typically unknown, making it challenging to define beforehand the specific area of focus for RRT exploration.

While previous studies have proposed next goal assignment strategies based on selecting points on the frontier, this study takes a novel approach by modifying the points on the frontier based on SDF information. Saulnier et al. [22] present an active exploration method in TSDF (Truncated Signed Distance Field) using Shannon mutual information. The method is specifically designed for robots equipped with RGB-D cameras. The goal assignment strategy of the robot focuses on reducing the entropy, which corresponds to minimizing uncertainty in the environment.

B. PATH PLANNING

Path planning can be classified as global and local path planning, with global path planning algorithms accounting for the entire environment, while local path planning only considers the immediate vicinity of the robot. The resulting global path provides the high-level guidance for navigation and is typically comprised of a sequence of local plans for execution. Dijkstra's and A* algorithms are two widely used graph search based methods for global path planning. The classical A* algorithm [2] is a variation of Dijkstra's algorithm that uses a combination of cost and heuristic function to evaluate and prioritize nodes to be expanded, allowing for faster convergence to an optimal solution than Dijkstra. In addition to these algorithms, the RRT (Rapidly-exploring Random Trees) [23] algorithm offers an alternative approach to global path planning. The RRT planner employs random sampling as its core strategy. It incrementally builds a tree structure by randomly sampled points in the configuration space and connecting them to the existing tree. This strategy allows the RRT algorithm to quickly explore the search space and find feasible paths. However, limitations of RRT are its inability to guarantee optimal paths, its sensitivity to the sampling distribution and the initial solution, as well as slow convergence to the optimal solution [24], often resulting in suboptimal solutions. To address these limitations, the RRT* (Rapidly-exploring Random Trees Star) algorithm was introduced [25]. RRT* enhances the original RRT algorithm by optimizing the paths found by the tree. After each new sample is added, the RRT* algorithm reconfigures the tree by rewiring its connections, considering the cost of alternative paths and minimizing the overall path cost. This enhancement allows RRT* to converge towards near-optimal solutions with increased iterations. Mashayekhih et al. [26] proposed informed planner based on RRT*, which shows lower computation cost with fewer iterations compared to RRT*. However, it's important to note that while RRT* improves the optimality of the paths, it does not guarantee the globally optimal path due to its reliance on random sampling. Additionally, RRT* primarily focuses on exploration and path optimization, with limited considerations for safety during the planning process. Wang et al. [24] proposed a planner combine RRT* with Convolutional Neural Network (CNN), which is trained using large amounts of successful paths planned by A* algorithm.

Furthermore, path planning can also be accomplished through optimization-based approaches, i.e., by treating the path planning problem as an Optimal Control Problem (OCP). OCP-based planning has exhibited notable performance in various applications, including unmanned surface vehicle navigation [27], autonomous flight deck path planning [28], and solutions for autonomous parking [29]. However, the efficiency and efficacy of OCP-based planners are contingent on the formulation of the optimization problem and the precision of the initial assumptions.

To discover an optimal path, numerical optimization techniques can be coupled with other methodologies such as the Artificial Potential Field (APF) [30]. APF is another global path planning method that relies on the creation of a virtual artificial potential field. However, APF has a known limitation related to local minima, which can render the target unreachable. To address this issue, several approaches have been proposed that combine APF with other algorithms. Keyu et al. [31] introduced a Fuzzy algorithm into APF to mitigate the problem of local minima. Additionally, the RRT algorithm has been used to assist APF in escaping local minima by selecting temporary goals [32]. Zhang et al. [33] enhanced the repulsive field of APF by introducing an additional force component to resolve the local minima challenge in the APF path planning for autonomous ships.

For local path planning, dynamic window approach (DWA) [34] and timed elastic band (TEB) [35] are commonly employed. The purpose of the local planning algorithm is to concurrently execute the tasks of tracking a moving target and avoiding obstacles. The DWA algorithm, which works as a velocity-based local planning algorithm, computes the optimal velocity necessary to reach the target while avoiding collisions. Due to its fast and effective control and obstacle avoidance capabilities, the DWA algorithm is used in our proposed method.

The Graph search-based methods such as A* [2] are widely applied as they look for the shortest path. However, the path planned by A* often approaches the edge of obstacles, which is unsafe in the autonomous exploration task facing unknown environment. To ensure a safe distance from obstacles, the inflation layer is widely used to generate a Costmap [6] [36], [37], [38], [39], [40], [41], [42], in which a buffer zone was reserved around obstacles to ensure a fixed safe distance from the robot, and the distance depends on the inflation radius of the inflation layer. While inflation layer aims to consider safety, it can also result in information loss, mission failure, and robot immobility in autonomous exploration tasks. For example, Yu et al. [7] mentioned that inflating the obstacles will prevent vehicles from planning feasible paths in narrow aisles. In addition, inflation radius in the Costmap needs to be determined based on the priori map, while during rescue or exploration, a priori map is not available. Zarrabi et al. [11] proposed a method to dynamically set the inflation radius parameter through the use of a Fuzzy controller, albeit at the cost of introducing a heavier computational load [10]. Some planning methods do not rely on Costmap, but maximize likelihood of reaching the goal [43] instead, in order to reduce the computational complexity at the cost of a path that is not the shortest. While this method enhances the safety of path planning to some degree, by avoiding narrow paths and favoring wider ones, it falls short in guaranteeing the robot's safety in cases where the destination is only accessible via narrow passages. Therefore, it is crucial that the robot's path planning module is equipped with safety design while being adaptable to different maps, to achieve safe autonomous exploration of unknown environments.

Distance maps have been used in path planning for collision checking, e.g., Sprunk et al. [44] employ a dynamically updated distance map of the environment, along with the robot's circumcircle and incircle, for efficient collision checks. Garimort et al. [45] apply similar method for humanoid robots. Oleynikova et al. [46] combine TSDF with Euclidean Signed Distance Field (ESDF) as a novel map representation for both 3D mapping and planning with image based sensing.

C. PATH SAFETY

Despite the importance of safety in path planning, the lack of a clear metric to quantify this factor has been noted [7], [40], [47], [48], [49], [50]. Some studies have reported a qualitative evaluation on safety, such as the absence of collisions [51]. However, considering localization error, information delay and other factors, the passing through the edge of obstacles (though not collided) is also extremely dangerous. Therefore, avoiding collisions alone does not guarantee the safety of robot. Several authors have proposed different metrics and definitions of safety. For example, Zhang et al. [43] define safety as the number of pathways the robot can take, as more pathways leaves more choices for obstacle avoidance. However, this definition is limited to scenarios with a wider path, and with no quantitative metrics on how wide it should be. Dang et al. [52] define a safety score specifically for RRT* planner used in aerial robots. Such a safety score quantified the distance between obstacles, and the local planner takes the safety score into account when planning flying route. Li et al. [49] employed the APF method to optimize the path segment near the vicinity of obstacles while simultaneously smoothing the path. Nevertheless, the APF method necessitates a priori knowledge of the map to ascertain the repulsive force exerted by obstacles, a requirement that is not usually met in autonomous exploration scenarios. Moreover, it also lacks a specific definition of the safety for the path generated by the 2D planners. While the existing literature provides valuable insights into safety in path planning for robots, current approaches often lack a standardized, quantitative measure of safety. To address this gap, the present study established safety metrics, which can easily quantify and compare the safety of different path planning modules under the same map.

D. MOTIVATION AND CONTRIBUTATION

The current challenges can be summarized as follows:

- Many researchers primarily focus on enhancing the efficiency of autonomous exploration, often at the expense of robot safety. Additionally, intelligent algorithms demand substantial computational resources, highlighting the urgent need for the development of a lightweight autonomous exploration solution that can ensure the safety of robots.
- Present research lacks a qualitative measure for assessing the safety of path planners. Engineering scenarios often involve factors like inaccurate localization and sensor errors, leaving less safety margins for path planning. Therefore, there is a pressing need for the definition of a universally applicable metric that can quantitatively evaluate the safety of the planned trajectories.
- Furthermore, the current direction of research on global path planning algorithm predominantly emphasizes reducing path length or incorporating intelligent method like CNN tailored for specific scenarios. However, few research focus on balancing robot safety and efficiency with adaptiveness across different scales of scenes. Hence, there is a demand for an adaptive path planning approach that prioritizes safety.

Accordingly, the key contributions of this work are as follows:

- to develop a safety-prioritized next goal assignment strategy in autonomous exploration that not only effectively increases the reachability of goals but also reduces the overlap of trajectory.
- to develop an SDF-based global path planner that enhances robot safety with high robustness and adaptability across different environmental scales without compromising path planning efficiency.

III. PROPOSED METHOD

Let $M \in R^2$ represent a bounded 2D grid map generated by Cartographer on which the autonomous exploration is performed. Each point $m = [x_m, y_m]^T \in M_0$ has an occupancy probability $p(m)$, where $p(m) = -1$ represents unknown area, $p(m) = 0$ means free, $p(m) = 100$ means occupied. The robot pose is represented by $\boldsymbol{x} = [x, y, \theta]^T$. In SDF map, the distance of every point to the nearest obstacle is known as $\boldsymbol{d}$. The path σ, which describes the robot's trajectory, is defined relative to the SDF map. It's important to note that the origin of this SDF map coincides with the origin of occupancy grid map.

The SDF map is shown in Fig. 2, where the black region represents the obstacle and the signed distance value for each point in the obstacle is negative ($d \leq 0$), while the non-black region represents the free space, with brighter region representing larger distance from obstacles, and the signed distance value for each point in the free space is positive ($d > 0$).

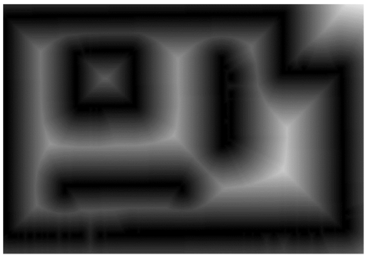


FIGURE 2. Demonstration of a signed distance field (SDF) Map, with the dark region representing obstacle area and bright region representing free space.

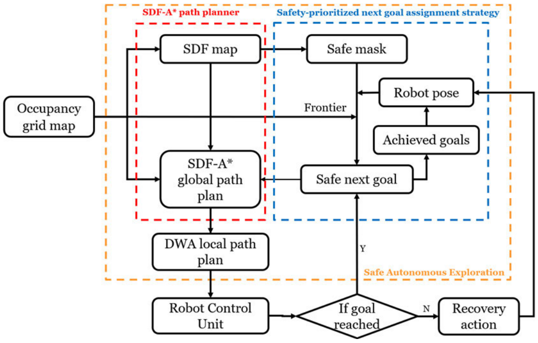


FIGURE 3. Overview of the proposed Safe Autonomous Exploration Strategies. The proposed method is comprised of SDF-A* planner and safety-prioritized next goal assignment strategy based on SDF map.

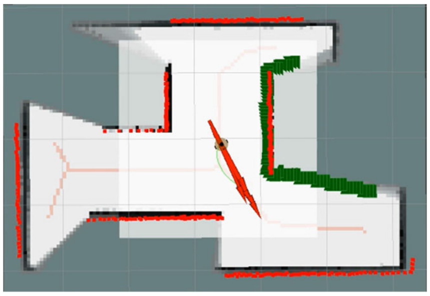


FIGURE 4. Visualization of continuous gap (green line) in Algorithm 1.

A. OVERVIEW

The proposed method, as depicted in Fig. 3, is comprised of SDF-A* planner and Safety-prioritized next goal assignment strategy based on an SDF map. The goal assignment publishes a safe goal to the path planning model using information from SDF map. Once the goal is reached through the path generated by SDF-A* (global path planner), DWA (local path planner) and robot control unit, a next goal will be assigned until there is no frontier in the map. If the robot fails to reach a goal, the recovery action is activated, which involves updating the robot pose and reassigning a new goal.

B. SDF-BASED SAFE AUTONOMOUS EXPLORATION STRATEGY

The proposed next goal assignment strategy aims to balance the safety and exploration efficiency by incorporating the distance from obstacles into the goal assignment process, ensuring that the assigned goal maintains a safe distance from obstacles in all environments without the need to adjust parameters of algorithm.

As the first step of next goal assignment, frontier detection and rough goal assignment is employed, as is described in Algorithm 1.

Algorithm 1, line 1: The function **findFrontiers** detects the frontiers from the 2D occupancy grid map from Cartographer based on Yamauchi's Frontier based approach [1];

Algorithm 1, line 2: F_m is the largest frontier, and a rough goal will be assigned on it through afterwards iteration.

Algorithm 1, line 6-18: To identify the occupancy condition of the frontiers according to the occupancy grid value, the continuous gap that is not an obstacle is searched for on **gap.**

To clarify, Algorithm 1 identifies continuous gaps in the environment based on occupancy grid values, not relying on the SDF map. In Fig. 4, the continuous gap is depicted as the green line, where the vertical portion corresponds to obstacles. Please note that the occupancy grid values are not 100 for now due to the sensor's delay in recognizing occupied grid cells as obstacles. The inclined section of the green line

Algorithm 1 Frontier Detection and Rough Goal Assignment

```
Input:   $M_0$ : Occupency Grid Map
         Pose of robot
Output:  Rough goal
1    $F_t$ = **findFroniers**($M_0$);
2    $F_m$ = **max**($F_t$);
     //Find Rough goal on the maximum Frontier
3    **if** is the first rough goal to find
4        **Rough goal** = Point on $F_m$ which is nearest to robot
5    **else**
6        **for** $* it = F_m.start() to * it = F_m.end()$
         //Find Gap on the maximum Frontier
7            **if** current *it is not obstacle and last*it is obstacle
8                gapStart=* it;
9            **else if** current * it is obstacle and last *it is not obstacle
10               gapEnd=*it;
11           **end if**
12           **if** gapDistance > $\kappa$ **then**
13               **Rough goal**= middle of gap on $F_m$
14               **break;**
15           **else**
16               go to line 7 to find new Gap;
17           **end if**
18       **end for**
19   **end if**
20   **return Rough goal**
```

Algorithm 2 Assign the Safe Goal

```
Input:   $M_0$ : Occupancy Grid Map
         Pose of robot
         Rough Goal: result from Algorithm 1
Output:  Safe goal
1    $M_t$ = Zhang − Suen thinning Map($M_0$);
2    safe mask = {$M_0$.size(), 0};
     //Find Safe goal using Depth First Search
3    for$i$ = 0,1,......$M_t$.size()
4      if $M_t$ (i) is free
5        safe mask(i) =1;
6        neighbour = DFS($M_t$ (i));
          //using DFS to find$M_t$ (i)′ sneighbour
7        for j in neighbour
8          if $M_t$ (j) is free && connected to space where safe
     mask =1
9            safe mask(j) =1;
10           delete j from neighbour;
11         else
12           safe mask(j) =1;
13           delete j from neighbour;
14         end if
15       end for
16     else
17       safe mask(i) = 0;
18     end if
19   end for
     //Find path from Rough goal to Pose of robot and refine the
     Rough goal to Safe goal when path touching the safe mask=1
20   p =Dijkstra findPath(Robot pose, Rough goal);
21   for Point x on p
22     if safe mask(x) == 1
23       Safe goal = x;
24       break;
25     else
26       x++;
27     end if
28   end for
29   return Safe goal
```

is frontiers. We initially assign the rough goal to the center of the gap and then refine this rough goal into a safe goal through Algorithm 2. The primary motivation for this refinement is to avoid potential issues where, during autonomous exploration, the occupancy values of some frontiers often fall between 0 and 100, and time is required to confirm whether a frontier is free or an obstacle. This precaution ensures that the assigned goals do not stuck the robot around obstacles, ultimately preventing failure in the autonomous exploration task.

Algorithm 2, line 1: Zhang-Suen method [53] is used to thin the occupancy grid map. Extracting the skeleton of the occupied map using the Zhang-Suen method can effectively reduce the occurrence of unreachable goals caused by the non-100 values of obstacles, and the performance depends on how often the map updates;

Algorithm 2, line 3-19: Depth-first search (DFS) method is used to find the connecting points that are furthest from the obstacle in the SDF map and store them in the **safe mask**. On the safe mask map, the path from the rough goal to robot is generated by Dijkstra algorithm to find the shortest path. During path production, the process starts from the rough goal, and the path is continuously expanded in all directions according to Dijkstra's algorithm to find the shortest path;

Algorithm 2, line 20-29: When the obtained shortest path intersects with the safe mask, the rough goal is optimized to the safe goal. Because the safe mask lines consist of the furthest points from the obstacle found by the SDF map, the safe goal on the safe mask can guarantee the safety of the autonomous exploration.

C. SDF-BASED A* GLOBAL PATH PLANNER

This algorithm fully considers the distance factor between the robot and the obstacle and establishes a distance function model. The function is designed to reach a balance among safety issue (distance from robot to obstacles), the 'Potential' (distance cost from the starting point to the robot's current position) and the 'Distance' (distance from the robot's current position to the goal). This balance is vital for optimizing the efficiency and safety of robotic navigation.

The SDF factor for a robot pose $\boldsymbol{x}$ that lies along a path $\boldsymbol{\sigma}$ can be expressed as follows:

$$s[m|x,\alpha] = \begin{cases} \dot{d} & if\ d > 0 \\ Inf & if\ d \leq 0 \end{cases} \tag{1}$$

Here $\dot{\boldsymbol{d}}$ represents the normalized result of derivative of d on the distance field. When $\boldsymbol{d} > \mathbf{0}, \boldsymbol{s} \in [\mathbf{0}, \mathbf{1}]$. We set $\boldsymbol{s}$ to $\boldsymbol{Inf}$ when $\boldsymbol{d} \leq \mathbf{0}$, to ensure robot does not consider points inside obstacles when planning the path.

The heuristic function of A* [23] is improved to account for the SDF factor as follows:

$$F(m) = P \times (\kappa_1 + s[m|x,\alpha]) + D(m)\times (\kappa_2 + s[m|x,\alpha]) \tag{2}$$

In the heuristic function $F(m)$ of the SDF-A* algorithm, P stands for Potential; D refers to Distance; κ_1 and κ_2 are coefficients that balance the weight of safety factor, Potential and Distance.

In our experiments, we set κ_1 and κ_2 to 1, indicating equal weight for Potential and Distance. However, it's essential to note that κ_1 and κ_2 can be set to different values, reflecting varying weightings for Potential and Distance. While κ_1 and κ_2 were both set to 1 in our simulations and real-world experiments, resulting in excellent performance, this idea originates from traditional A* planner, where the weightings assigned to Potential and Distance are typically equal.

SDF factor is added to balance the trade-off between safety and path cost. Instead of directly using distance values in SDF, this study employs spatial derivative of distances to make the algorithm more adaptive to different scenarios. For example, the distance of different areas on the map from the obstacle may vary greatly, from the centimeter level in the narrow passages to the meter level in big warehouse, if the distance values of SDF map are used directly, the heuristic function $F(m)$ will vary too much and the comparison of $F(m)$ will be meaningless. Therefore, the distance derivative values $\dot{d}$ is used in the current path planning model to account for the relative distance, thus avoiding the need to adjust parameters like radius of inflation layer in the traditional method.

IV. EXPERIMENTS

The performance of the proposed SDF-A* path planning method and SDF-based Safe Autonomous Exploration Strategy were first evaluated through simulations in ROS, and then tested in real-world environments using a robot equipped with RPLIDAR-A3 single-line lidar. Furthermore, quantitative comparisons were also made in the following section.

A. EXPERIMENTS IN SIMULATION ENVIRONMENT

The performance of the proposed SDF-A* path planning method and SDF-based Safe Autonomous Exploration Strategy were evaluated through simulations in ROS. A cylindrical robot with a diameter of 0.3m and a height of 0.65m, equipped with lidar is used as the mobile robot platform. To make the simulation process more realistic, the lidar parameter were set the same as RPLIDAR-A3 lidar, which is used in real-world experiments. The computer operating system is Ubuntu 18.04, and two-dimensional grid map is displayed on Rviz.

For path planning simulation, a 6 m × 4 m environment was constructed in the Gazebo environment, and 4 goals are assigned from the map generated by Cartographer as shown in Fig. 5. Both wide and narrow channels were included in the map, to allow for a fair comparison of the performance of different path planning algorithms. The start and endpoints, as well as the robot's pose, are fixed for each segment. Additionally, in order to evaluate the impact of factors such as environment size and the number of obstacles on the path planning module, we have also simulated a 391 m^2 environment with simple obstacles and complex obstacles, as shown in Fig. 6(a) and 6(b), respectively. The environment with simple obstacles has 12 fixed goal points, while the environment with complex obstacles has 18 fixed goal points, as shown in red points in Fig. 6.

Three types of global path planners were evaluated in this study, including 1). the baseline approach, which combines ROS's move_base navigation package using an A* algorithm for global planning; 2). RRT*; 3). SDF-A* for global planning proposed in this paper. All approaches apply DWA for local path plan.

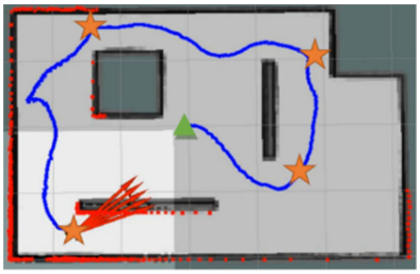


FIGURE 5. Simulation environment for the comparison of different path planning algorithms, with the starting point fixed at the center of the map (green triangle) and four fixed goals shown as orange stars.

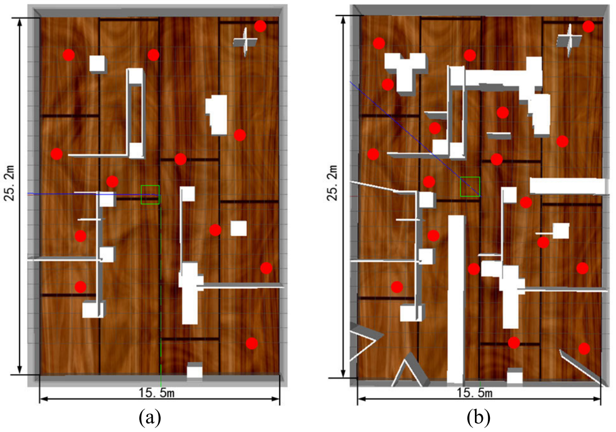


FIGURE 6. Simulation environment in a 391 m^2 environment with (a) fewer obstacles and 12 goals; (b) with more obstacles and 18 goals. This simulation is established to assess the impact of environment size, obstacle quantity, and complexity on the performance of path planning module. Fixed goals are shown in red points.

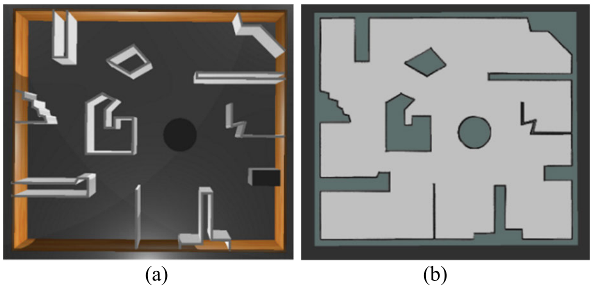


FIGURE 7. Simulation environment for autonomous exploration based on Gazebo (a) and Rviz (b).

For autonomous exploration, we compared three exploration strategies: the classic frontier detector, the WFD, and the SDF-based Safe Autonomous Exploration Strategy in a simulation platform (around 247 m^2) as shown in Fig. 7.

B. EXPERIMENTS IN REAL-WORLD ENVIRONMENT

The proposed strategy is implemented on the three-wheeled robot (two wheeled differentially placed in rear, one single free wheel additional placed in front to ensure the robot equilibrium) equipped with Jetson Nano and RPLIDAR-A3 lidar. The lidar has a maximum detecting range of 25m, and 6rad angle, with 1° resolution. An i7 embedded computer carries out all onboard processing. The robot is equipped with two active wheels, each having a diameter of 0.65 meters, and one passive wheel. The mobile robot had a physical size of 0.25 m × 0.18 m × 0.11 m, a maximum linear velocity of 0.2 m/s, a maximum angular velocity of 0.3 rad/s, a maximum linear acceleration of 0.15 m/s, and a maximum angular acceleration of 0.25 rad/s.

The experimental evaluation of the proposed SDF-based Safe Autonomous Exploration Strategy was tested on a robot equipped with the SDF-A* planner and RPLIDAR-A3 sensor for perception and navigation. The chosen actual scene represents typical indoor environments with obstacles, narrow passages, and varying complexities to mimic real-world scenarios. The proposed method is implemented as ROS package on Jetson Nano and tested in two indoor environments (scene 1 around 30m^2, scene 2 around 120 m^2), as is shown in Fig. 14(a) and Fig. 15 (a).

C. EVALUATION METRICS

The path planning algorithms are evaluated using the following metrics:

- Path Length: the length of the path is calculated using the robot's pose data published by Cartographer;
- Time Consumption: the time consumed from starting point to endpoint;
- Path Safety: as discussed above, since there is no quantitative metric for path security, we design a statistical metric to measure path security based on SDF.

The trajectory is extracted from mapping result in Rviz using OpenCV, as shown in Fig. 8 (a). The path is then discretized into hundreds of points evenly spaced along the curve, denoted as $p_i = (x_i, y_i)$. The safety of the path is evaluated based on the SDF values of the points on the path, as shown in Fig. 8 (b). Specifically, we calculate the SDF value d_i for each point p_i on the path using the following equation:

$$d_i = SDF(x_i, y_i) \tag{3}$$

where $SDF(x_i, y_i)$ is the SDF value of the corresponding location (x_i, y_i) on the map. The safety of the path is then calculated as the average of all SDF values of the points, which is computed as follows:

$$Path\ safety = \frac{1}{N} \times \sum\nolimits_i (d_i) \tag{4}$$

where N is the total number of points on the path.

Larger average SDF value indicates greater distance from the obstacle, thus represents a safer path. So this metric can effectively reflect the safety of paths with the same start-end point for different path planning strategies.

V. RESULTS AND DISCUSSION

A. PERFORMANCE OF SDF-A* PATH PLANNER

1) RESULTS OF PATH PLANNERS' EXPERIMENTS

This study runs 10 repetitive experiments on simulation using A* planner, RRT* planner and the proposed SDF-A* planner on three maps with the fixed start and end points, as is shown in Fig. 5 and Fig. 6. The average total time consumption, total path length and path safety is recorded, as is shown in Fig. 9.

It can be seen that, under these three conditions, the utilization of the SDF-A* path planner has resulted in a 30.10% improvement in path safety, a 64.11% reduction in time, and a 31.67% decrease in path length compared to the A* planner in small environment, and 12.63%, 10.29%, 6.15% respectively in large simple environment, 7.86%, 13.26%, 9.52% respectively in large complex environment.

In comparison to the RRT* planner, the SDF-A* outperforms in terms of path safety, achieving a 10.59% improvement in small environment, 2.59% in large simple environment and 10.62% in large complex environment. The reduction in time consumption with SDF-A* is 34.65%, 7.33%, and 22.06% in the three scenarios, and the reduction in path length is 18.07%, 13.80%, and 9.90%, respectively.

These results indicate that the SDF-A* planner outperforms the classical A* planner and RRT* planner in terms of path length, time consumption and efficiency under both small and large environment, with various settings of obstacles.

2) DISCUSSIONS ON PATH PLANNERS

Fig. 10 shows the partial SDF map along with the corresponding robot trajectories using A* planner and SDF-A* planner. It can be seen that, incorporating information from the SDF results in shorter path with less time, although the path length planned by SDF-A* is not the shortest. Fig. 10(b) illustrates the actual trajectory of the robot under SDF-A*, which aligns with the white space shown in Fig. 10(a), representing the furthest point from the obstacle in the SDF map. This alignment demonstrates that the SDF information has a significant impact on the actual robot trajectory.

Although the A* path planning algorithm is widely recognized for generating the shortest path between a robot's initial and goal locations, a drawback of A* arises when its shortest path intersects with the edge of the inflation radius, which can compel the robot to decelerate as it approaches the edge or even reverse, as illustrated by the yellow circles in Fig. 10(c). This unnecessary reversing and deceleration can significantly impact the actual performance of the A* path planner, particularly in narrow paths, leading to trajectories that are not as short as expected.

It is indeed possible that the SDF-A* planned path, as indicated by the green line in Fig. 11(b), can be longer than the shortest path, as computed by the traditional A* planner in Fig. 11(a). However, as is shown in Fig. 11(c), a longer trajectory was found due to the deceleration within the inflation radius, i.e., the robot continuously reverses and stops within the inflation radius when following A* planned path.

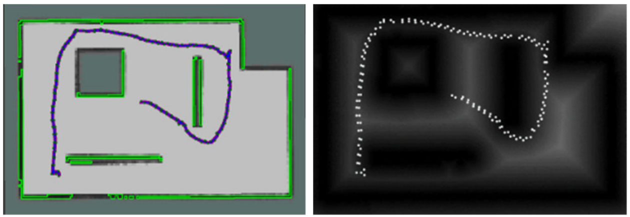
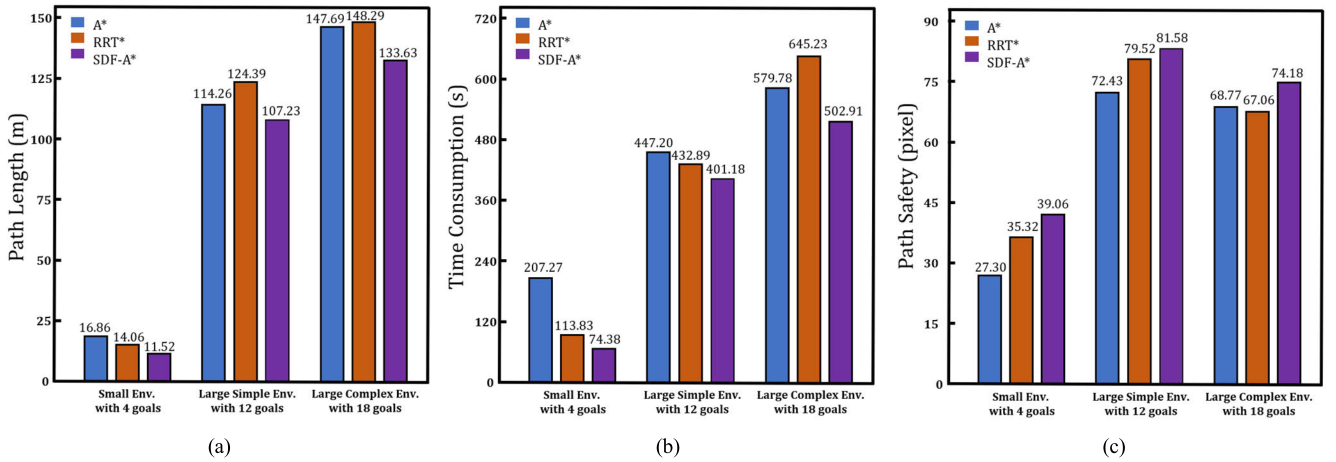
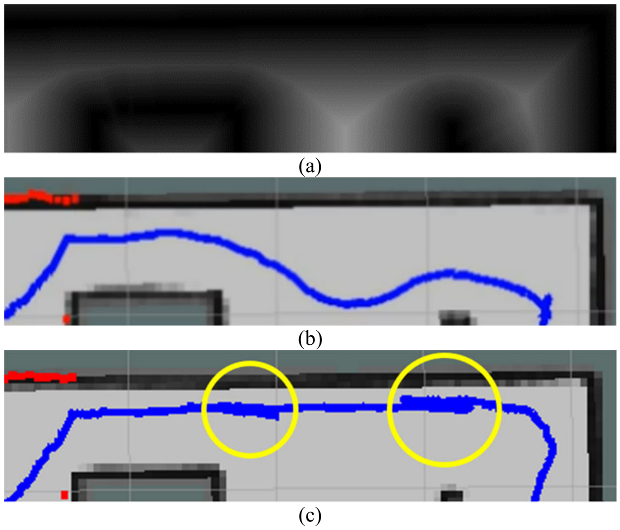


FIGURE 8. Extract path trajectory (purple line) in an occupancy grid map (a)), then evenly discretized the trajectory into 150 points and put the points (white points) in the SDF map (b).



(a) (b) (c)

FIGURE 9. Comparison of three path planning algorithms under different environments, with the results averaged from 10 repetitive experiments in three environments.

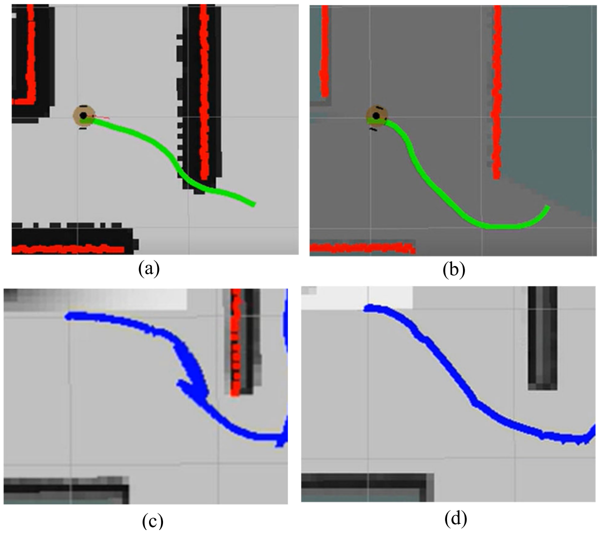


(a)

(b)

(c)

FIGURE 10. (a) A partial representation of the SDF map, where the white region is the furthest points from obstacle. (b)Robot trajectory generated by SDF-A* planner. (c) Robot trajectory generated by A* planner with obvious reversing and turning.



(a) (b)

(c) (d)

FIGURE 11. Path planned by (a) A*, and (b) SDF-A* in green lines, and actual trajectory by (c) A* with reverse and by (d) SDF-A* without inefficient behaviors.

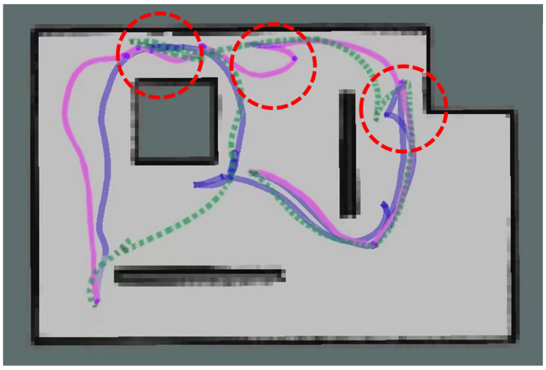


FIGURE 12. Representative trajectories planned by RRT* and unnecessary turns caused by randomness indicated by red dashed circles.

On the other hand, while the trajectory obtained with the SDF-A* planner in Fig. 11(d) has a longer planned path, a shorter actual trajectory was resulted due to the avoidance of inefficient behaviors like reversing and getting stuck.

It can also be seen from Fig. 9 that, the benefits of SDF-A* is more noticeable under environment with more obstacles. This is because the increased obstacle density reduces the average distance between the robot and obstacles, posing more challenges to A* and RRT* algorithms. However, in larger environments with fewer obstacles, the improvement in path safety of SDF-A* was relatively small. This is because environments with sparse obstacles posed less challenge to all path planners.

Furthermore, the RRT* planner occasionally generates paths that deviate significantly from the optimal route, due to its randomly sampling nature, as shown in Fig. 12. The randomness occasionally leads to larger directional changes

TABLE 1. Comparison of three different exploration strategies, with data averaged from 10 repetitive experiments.

Exploration strategy	Exploration Time (s)	Trajectory length(m)	Goal number	Safety(pixel)	Coverage	Failure rate
Classic Frontier Detection	600	103.78	46	45.00	90.6%	50%
WFD	762	101.08	22	61.82	86.78%	30%
SDF-Safe Exploration	421	87.88	25	66.18	92.56%	0%

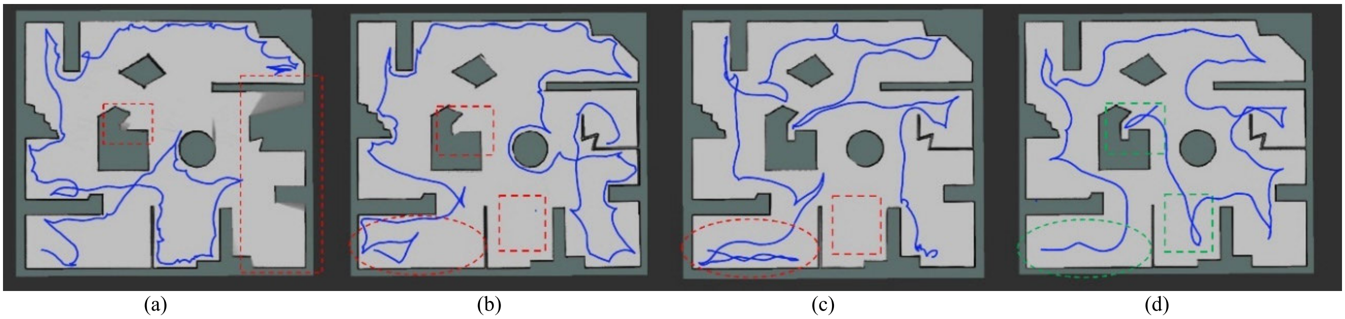


FIGURE 13. Simulation results of autonomous exploration strategy using (a) and (b) frontier-based exploration strategy, (c) WFD strategy and (d) SDF-based Safe Autonomous Exploration Strategy. The robot applied frontier-based exploration and WFD ends up with stuckness near obstacle, or fails to cover certain area, as shown in red dashed rectangle, and the unnecessary overlap of trajectories is shown in red dashed circles in (b) and (c). In contrast, the proposed strategy demonstrates enhanced exploration efficiency, with no unnecessary trajectory overlaps (depicted by green dashed circles and rectangles in (d)).

and unnecessary turns, as indicated by the red circles in Fig. 12, which highlight that the RRT* algorithm, with an improvement over RRT, still exhibits a considerable degree of randomness in the paths it generates.

B. PERFORMANCE OF SDF-BASED SAFE AUTONOMOUS EXPLORATION STRATEGY

1) RESULTS OF SIMULATION EXPERIMENTS

To evaluate the performance of the autonomous exploration strategies, the SDF-based Safe Autonomous Exploration Strategy was combined with the SDF-A* planner, and its performance was compared with the classical frontier-based strategy and WFD-exploration strategy. The experiments were repeated 10 times in a simulated environment of 247.06 m^2, as is shown in Fig. 7.

The experimental results are shown in Fig. 9. The classic frontier-based exploration strategy exhibited some limitations, resulting in exploration failures in 5 out of 10 experiments. The robot's trajectory length was relatively long (103.78m) due to its tendency to select goals close to obstacles, leading to safety risk (45.00 pixels). The strategy employed 46 goals, achieving a coverage of 90.6%, but the high failure rate (50%) indicated its susceptibility to getting stuck in narrow passages and incomplete map coverage, as shown in red dashed rectangles in Fig. 13 (a) and (b).

The WFD frontier-based exploration strategy showed a moderate performance, experiencing exploration failures in 3 out of 10 experiments. The strategy selected fewer average goals (22) as the strategy sometimes ignored smaller frontiers, resulting in a shorter trajectory length (101.08 m) compared to the classic approach. However, the safety score was compromised (61.82 pixels) and the coverage achieved was 86.78%, indicating reasonable but not complete exploration. A representative exploration path using this method is shown in Fig. 13(c).

The proposed SDF-based Safe Autonomous Exploration Strategy outperformed the other two strategies, successfully completing the exploration task in all 10 experiments without any failures. The strategy achieved a high level of safety (66.18 pixels) by efficiently avoiding unnecessary path overlaps, as shown in Fig. 13(d). It selected 25 goals on average, resulting in a relatively shorter trajectory length (87.88 m) and a higher coverage ratio of 92.56%. The application of this approach resulted in a substantial 47.06% improvement in path safety, a significant 44.75% reduction in time, and a notable 15.32% decrease in path length when compared to the traditional frontier-based exploration strategy.

2) DISCUSSIONS ON SIMULATION RESULTS

Through ten repetitive experiments, our method consistently achieved high coverage rates and shorter trajectory lengths for several reasons. The traditional frontier-based exploration and WFD strategies often fall short in covering specific areas, due to their lack of consideration of the safety of the next goal, leading the robot towards an unreachable destination and being stuck, resulting in unexplored regions on the map. Our method, on the other hand, employs the SDF map and safe mask in Algorithm 2 to optimize the selection of the next goal, ensuring that the robot does not get stuck due to an unsafe goal.

Additionally, the trajectories generated by traditional frontier-based exploration and WFD frequently exhibit unnecessary overlaps, increasing the overall trajectory length.

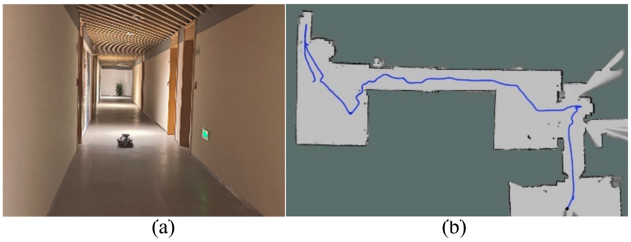


FIGURE 14. Indoor experiments in scene 1 (around 30m^2). (a) the actual environment. (b) Occupancy grid map with the blue line showing autonomous exploration trajectory and black line showing the contour of the resulted map.

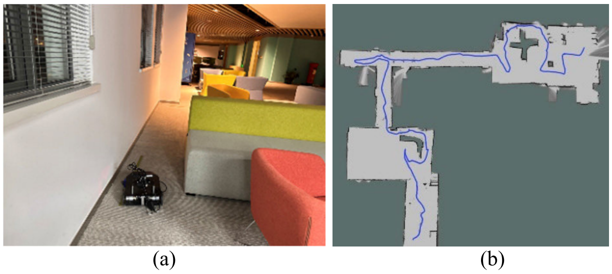


FIGURE 15. Indoor experiments in scene 2 (around 120 m^2). (a) the actual environment. (b) Occupancy grid map with the blue line showing autonomous exploration trajectory and black line showing the contour of the resulted map.

Our approach resolves this issue by optimizing the next goal selection strategy. By recording the robot's previous path and assessing the likelihood of overlap with already traversed trajectories, we enhance the efficiency of the path planning process, ultimately achieving a combination of high coverage rates and shorter trajectory lengths.

3) RESULTS OF REAL-WORLD EXPERIMENTS

The proposed SDF-A* combined with SDF-based Safe Autonomous Exploration Strategy was also implemented on a robot and tested in real-world indoor environment. It can be seen from Fig. 14 that, in an environment with long corridor, the autonomous exploration process took around 767 s in scene 1 with 21 goals assigned, which results in a path length of 13.98m. Similarly, in Scene 2 (Fig. 15), characterized by narrow passages, office sofas, and chairs, the proposed method effectively explored the unknown indoor area. In this scenario, the exploration process took about 964 seconds, covering a trajectory length of 24.8 meters, with 59 goals assigned. The implementation of the proposed method has shown successful and safe attempts in exploring the indoor unknown area.

VI. CONCLUSION

This paper presents a novel approach based on signed distance field to optimize autonomous exploration strategies and path planning algorithms for safe, effective, and adaptive navigation for indoor rescue and search missions. A quantitative criterion is established to evaluate the safety and the probability of being trapped for path planning.

The proposed SDF-based Safe Autonomous Exploration Strategy and SDF-A* path planner outperform traditional methods, as demonstrated through experiments on simulated robots. Specifically, the SDF-A* path planner has led to a 30.10% increase in path safety and a 64.11% reduction in time, as well as a 31.67% reduction in path length, compared with A* planner. Furthermore, the SDF-A* path planner can effectively prevent robots from unnecessarily reversing, slowing down, or getting trapped within the inflation radius. The SDF-A* path planner also has the advantage of being self-adaptive to environment, without prior parameter setting. This is important because it allows the proposed method to adapt to different types of environments and map configurations without the need for human intervention.

The SDF-based Safe Autonomous Exploration Strategy combined with SDF-A* path planner has led to a 47.06% increase in path safety and a 44.75% reduction in time, as well as a 15.32% decrease in path length compared with traditional frontier-based exploration strategy. Additionally, the coverage ratio is 5.78% higher than the traditional frontier-based method. Most importantly, the proposed approach results in a 0% failure rate, while that of frontier-based method and WFD method is 50% and 30%, respectively. Although the SDF-based approach requires more goals to complete the exploration task, it exhibits higher success rate, higher degree of safety, and lower time consumption than the traditional frontier-based approach.

Real-world experiments further validate the feasibility, efficiency, and safety of the proposed methods for indoor rescue and search missions. The performance improvements attained by the SDF-based Safe Autonomous Exploration Strategy and SDF-A* path planner highlight their potential for practical implementation in real-world scenarios, contributing to more effective and reliable robotic missions aimed at saving lives and property during emergencies.

However, there are some limitations that arise when the algorithm incorporates the SDF map.

1. The SDF-A* planner requires the generation of SDF map, which introduces additional computational cost. However, it's worth noting that in our experiments, which include simulated environments ranging from 24 m^2 to 247 m^2 and real-world settings spanning from 30 m^2 to 120 m^2, we did not observe any significant computational overload on robot navigation.

2. The SDF-based Safe Autonomous Exploration Strategy may not offer substantial efficiency improvements in environments with vast, nearly obstacle-free spaces. This is because SDF maps under such environment lacks obstacles, tend to exhibit nearly uniform characteristics, rendering them less effective in the current algorithm.

REFERENCES

[1] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang, and Y. W. Yun, "An improved frontier-based approach for autonomous exploration," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 292–297.

[2] P. Hart, N. 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.

[3] J. Orsulic, D. Miklic, and Z. Kovacic, "Efficient dense frontier detection for 2-D graph SLAM based on occupancy grid submaps," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3569–3576, Oct. 2019.

[4] L. Pang, J. Hu, P. Xiao, and S. Liu, "Active SLAM based on geometry rules and forward simulation in exploration space," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2018, pp. 238–243.

[5] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1396–1402.

[6] J. A. Placed and J. A. Castellanos, "Fast autonomous robotic exploration using the underlying graph structure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 6672–6679.

[7] J. Yu, J. Hou, and G. Chen, "Improved safety-first A-star algorithm for autonomous vehicles," in *Proc. 5th Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Dec. 2020, pp. 706–710.

[8] S. Macenski, T. Moore, D. V. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2," *Robot. Auto. Syst.*, vol. 168, Oct. 2023, Art. no. 104493.

[9] C. S. Chen, C. J. Lin, C. C. Lai, and S. Y. Lin, "Velocity estimation and cost map generation for dynamic obstacle avoidance of ROS based AMR," *Machines*, vol. 10, no. 7, p. 501, Jun. 2022.

[10] R. Yuan, F. Zhang, J. Qu, G. Li, and Y. Fu, "A novel obstacle avoidance method based on multi-information inflation map," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 47, no. 2, pp. 253–265, Aug. 2019.

[11] N. Zarrabi, R. Fesharakifard, and M. B. Menhaj, "Dynamic reconfiguration of costmap parameters with fuzzy controllers to enhance path planning," in *Proc. 10th RSI Int. Conf. Robot. Mechatronics (ICRoM)*, Nov. 2022, pp. 427–432.

[12] Z. Zheng, S. He, and J. Pan, "Efficient exploration in crowds by coupling navigation controller and exploration planner," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12126–12133, Oct. 2022.

[13] H. Huang, Y. Sun, H. Ye, and M. Liu, "Metric monocular localization using signed distance fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Macau, Nov. 2019, pp. 1195–1201.

[14] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1366–1373.

[15] M. N. Finean, W. Merkt, and I. Havoutis, "Simultaneous scene reconstruction and whole-body motion planning for safe operation in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3710–3717.

[16] J.-D. Fossel, K. Tuyls, and J. Sturm, "2D-SDF-SLAM: A signed distance function based SLAM frontend for laser scanners," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1949–1955.

[17] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 215–236, Feb. 2014.

[18] P. G. C. N. Senarathne and D. Wang, "Incremental algorithms for safe and reachable frontier detection for robot exploration," *Robot. Auto. Syst.*, vol. 72, pp. 189–206, Oct. 2015.

[19] Z. Sun, B. Wu, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Frontier detection and reachability analysis for efficient 2D graph-SLAM based active exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2051–2058.

[20] L. Zhao, L. Yan, X. Hu, J. Yuan, and Z. Liu, "Efficient and high path quality autonomous exploration and trajectory planning of UAV in an unknown environment," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 10, p. 631, Sep. 2021.

[21] B. P. L. Lau, B. J. Y. Ong, L. K. Y. Loh, R. Liu, C. Yuen, G. S. Soh, and U.-X. Tan, "Multi-AGV's temporal memory-based RRT exploration in unknown environment," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9256–9263, Oct. 2022.

[22] K. Saulnier, N. Atanasov, G. J. Pappas, and V. Kumar, "Information theoretic active exploration in signed distance fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4080–4085.

[23] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep., 1998.

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

[26] 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.

[27] X. Wang, Z. Deng, H. Peng, L. Wang, Y. Wang, L. Tao, C. Lu, and Z. Peng, "Autonomous docking trajectory optimization for unmanned surface vehicle: A hierarchical method," *Ocean Eng.*, vol. 279, Jul. 2023, Art. no. 114156.

[28] X. Wang, B. Li, X. Su, H. Peng, L. Wang, C. Lu, and C. Wang, "Autonomous dispatch trajectory planning on flight deck: A search-resampling-optimization framework," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105792.

[29] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 4327–4332.

[30] H. Li, W. Liu, C. Yang, W. Wang, T. Qie, and C. Xiang, "An optimization-based path planning approach for autonomous vehicles using the DynEFWA-artificial potential field," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 2, pp. 263–272, Jun. 2022.

[31] L. Keyu, L. Yonggen, and Z. Yanchi, "Dynamic obstacle avoidance path planning of UAV based on improved APF," in *Proc. 5th Int. Conf. Commun., Image Signal Process. (CCISP)*, Nov. 2020, pp. 159–163.

[32] H. An, J. Hu, and P. Lou, "Obstacle avoidance path planning based on improved APF and RRT," in *Proc. 4th Int. Conf. Adv. Electron. Mater., Comput. Softw. Eng. (AEMCSE)*, Mar. 2021, pp. 1028–1032.

[33] L. Zhang, J. Mou, P. Chen, and M. Li, "Path planning for autonomous ships: A hybrid approach based on improved APF and modified VO methods," *J. Mar. Sci. Eng.*, vol. 9, no. 7, p. 761, Jul. 2021.

[34] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[35] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, "Efficient trajectory optimization using a sparse model," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2013, pp. 138–143.

[36] Z. Cheng, B. Li, and B. Liu, "Research on path planning of mobile robot based on dynamic environment," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2022, pp. 140–145.

[37] P. Marin-Plaza, A. Hussein, D. Martin, and A. D. L. Escalera, "Global and local path planning study in a ROS-based research platform for autonomous vehicles," *J. Adv. Transp.*, vol. 2018, pp. 1–10, Jan. 2018.

[38] M. Kobayashi and N. Motoi, "Path planning method considering blind spots based on ROS navigation stack and dynamic window approach for wheeled mobile robot," in *Proc. Int. Power Electron. Conf.*, May 2022, pp. 274–279.

[39] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 709–715.

[40] C. Wang, J. Wang, C. Li, D. Ho, J. Cheng, T. Yan, L. Meng, and M. Q.-H. Meng, "Safe and robust mobile robot navigation in uneven indoor environments," *Sensors*, vol. 19, no. 13, p. 2993, Jul. 2019.

[41] J. Zhao, S. Liu, and J. Li, "Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS," *Sensors*, vol. 22, no. 11, p. 4172, May 2022.

[42] S. Jiang, S. Wang, Z. Yi, M. Zhang, and X. Lv, "Autonomous navigation system of greenhouse mobile robot based on 3D LiDAR and 2D LiDAR SLAM," *Frontiers Plant Sci.*, vol. 13, Mar. 2022, Art. no. 815218.

[43] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Maximum likelihood path planning for fast aerial maneuvers and collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2805–2812.

[44] C. Sprunk, B. Lau, P. Pfaffz, and W. Burgard, "Online generation of kinodynamic trajectories for non-circular omnidirectional robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 72–77.

[45] J. Garimort, A. Hornung, and M. Bennewitz, "Humanoid navigation with dynamic footstep plans," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3982–3987.

[46] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *Proc. Workshop, Geometry Beyond-Represent., Phys., Scene Understand. Robot.*, 2016, pp. 1–7.

[47] E. Shang, B. Dai, Y. Nie, Q. Zhu, L. Xiao, and D. Zhao, "A guideline and key-point based A-star path planning algorithm for autonomous land vehicles," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[48] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, R. Siegwart, M. Hutter, and K. Alexis, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 3306–3313.

[49] H. Li, T. Zhao, and S. Dian, "Forward search optimization and subgoal-based hybrid path planning to shorten and smooth global path for mobile robots," *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 110034.

[50] M. Cao, B. Li, and M. Shi, "The dynamic path planning of indoor robot fusing B-spline and improved anytime repairing A* algorithm," *IEEE Access*, vol. 11, pp. 92416–92423, 2023.

[51] Z. Xiong, J. Eappen, A. H. Qureshi, and S. Jagannathan, "Model-free neural Lyapunov control for safe robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 5572–5579.

[52] T. Dang, S. Khattak, F. Mascarich, and K. Alexis, "Explore locally, plan globally: A path planning framework for autonomous robotic exploration in subterranean environments," in *Proc. 19th Int. Conf. Adv. Robot. (ICAR)*, Dec. 2019, pp. 9–16.

[53] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.

HEYING WANG was born in Changchun, Jilin, China, in 1997. She received the bachelor's degree from Tongji University. She is currently pursuing the master's degree with Shanghai Jiao Tong University. Her main research interests include path planning and autonomous exploration of mobile robots.

YUAN LIN was born in Harbin, Heilongjiang, China, in 1998. He received the bachelor's degree from Tongji University. He is currently pursuing the master's degree with Shanghai Jiao Tong University. His research interest includes simultaneous localization and mapping (SLAM).

WEI ZHANG is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University. His research interests include deep learning, laser diagnostics, and image processing.

WENTAO YE was born in Shanghai, China, in 2000. He is currently with the China-U.K. Low Carbon College, Shanghai Jiao Tong University. His research interests include preoperative and intraoperative image registration based on image feature information and intraoperative navigation of medical robotic arms.

MINGMING ZHANG received the B.S. degree in mechanical engineering from East China University, China, in 2015, and the M.S. degree in mechanical engineering and the Ph.D. degree in control science from Shanghai Jiao Tong University, China, in 2019 and 2022, respectively. He is currently a Postdoctoral Researcher with the Global Institute of Future Technology, Shanghai Jiao Tong University. His research interests include power conversion systems, model predictive control, and motion control.

XUE DONG received the bachelor's degree in thermal energy and power engineering from North China Electric Power University, Beijing, in 2011, and the Ph.D. degree in chemical engineering from The University of Adelaide, Australia, in February 2016. Since January 2021, she has been an Associate Professor with Shanghai Jiao Tong University. Her research interest includes multifaceted, encompassing areas such as laser testing technology, machine vision, and sensor fusion.