Robotica (2018) volume 36, pp. 96–118. © Cambridge University Press 2017
doi:10.1017/S0263574717000194

A safe area search and map building algorithm for a wheeled mobile robot in complex unknown cluttered environments

Andrey V. Savkin and Hang Li*

School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia

(Accepted March 18, 2017. First published online: April 10, 2017)

SUMMARY
In this paper, a safe map building and area search algorithm for a mobile robot in a closed unknown environment with obstacles is presented. A range finder sensor is used to detect the environment. The objective is to perform a complete search of the environment and build a complete map of it while avoiding collision with the obstacles. The developed robot navigation algorithm is randomized. It is proved that with probability 1 the robot completes its task in a finite time. Computer simulations and experiments with a real Pioneer-3DX robot confirm the performance of the proposed method.

KEYWORDS: Area search; Mapping; Obstacle avoidance; Safe robot navigation; Unknown cluttered environments; Wheeled mobile robots; Robotic search and rescue.

1. Introduction

Area search and map building in cluttered environments is a fundamental problem of modern robotics. In many military and commercial applications of mobile robots, the robot should be able to explore an unknown area with obstacles and build a geometrical map including the location and shape of each obstacle. The general application of the mobile robot is searching and mapping a dangerous environment which is impossible or undesirable to reach by humans. Possible examples include search and rescue operations in disaster areas, military intelligence, surveillance of contaminated areas, marine archaeology, and exploration of other planets. Moreover, similar problems appear in many applications of service robotics such as navigation of industrial robots, agricultural robots,[42] and hospital robots.[38] Various area search and map building problems for mobile robots have attracted a lot of attention in the robotics community; see, e.g., refs. [3, 15, 17, 21, 25, 27, 28, 31, 37, 39, 41]. Recent publications in this field present many achievements in both single robot mapping and multi-robot mapping. The frontier-based exploration is the most common method in single robot exploration and mapping; see, e.g., refs. [2, 12, 29, 39]. For multi-robot systems, not only frontier-based algorithms (see, e.g., refs. [1, 8]) but other approaches (see, e.g., refs. [13, 20]) were proposed. In this paper, we focus on the single robot exploration. The work in ref. [2] proposed an efficient and simple algorithm to explore a closed environment by using incremental triangulation. However, this algorithm requires the obstacles in the environments are polygons. Another work in ref. [29] presented a fast exploration algorithm based on a simple prior topological map of the environment. One of the disadvantages is that the algorithm of ref. [29] cannot be implemented in a completely unknown environment. The works in refs. [12, 39] proposed two similar frontier-based exploration algorithms with search trees. The main difference of these two algorithms is the algorithm of ref. [12] selects a candidate randomly and the algorithm of ref. [39] determines the candidate by a multi-criteria decision method.

Another fundamental problem of robotics is safe (collision free) navigation of a mobile robot in cluttered environments with obstacles. In order to operate in a cluttered unknown environment, a mobile robot should be able to detect and avoid obstacles along its trajectory. This problem has been

* Corresponding Author. E-mail:hang.li1@student.unsw.edu.au

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

the subject of extensive research in robotics over decades; see, e.g., refs. [14, 23, 34] and references therein. There are many achievements in recent publications in this field; e.g., refs. [4, 18, 30]. The work in ref. [4] improved and modified Vector Field Histogram algorithms based on range finder sensor for a non-static environment. The most important modification of this work appears in the calculation of histogram values, which takes into account the usage of laser range finder (LRF). In the work in ref. [18], a heuristic obstacle avoidance algorithm with a camera sensor was proposed. According to the comparison in ref. [18], the algorithm of ref. [18] has a shorter navigation trajectory than some previous works. However, the performance of the algorithms in both ref. [4] and ref. [18] has not been proved by mathematically rigorous analysis. Moreover, these two methods do not take into account the constraints of robot's motion. The work in ref. [30] presented a safe navigation method with a 360° camera, which is different from the type of the camera used in ref. [18]. However, the same limitation of the methods of ref. [30] and ref. [18] is that the specific features (e.g., line segment or corner) in the environments are required to calculate the collision-free regions. A number of algorithms for robot navigation with obstacle avoidance for non-holonomic robots was proposed in refs. [14, 23, 34]. All these publications do not consider the problem of map building.

In this paper, these two fundamental problems of modern robotics are combined together and a robot navigation algorithm for exploring a complex unknown environment with obstacles and building a binary map while avoiding collisions with obstacles is proposed. The task is performed by a wheeled mobile robot equipped with a range finder sensor. The search area is cluttered with a number of possibly non-convex obstacles with unknown shapes and locations. An algorithm for robot collision free navigation and map building is developed to generate a complete map of the searched area based on the range finder sensor measurements. It is proved that with probability 1 the task of the area search and map building will be completed in a finite time.

The main feature of our method is that it combines complete map building with obstacle avoidance for an unicycle model with constant speed, whereas some other papers did not solve the safe navigation problem for this model. For example, the works in refs. [12, 39] proposed different methods to select the candidate points and build a search tree. However, like the most of other frontier-based exploration methods, these methods did not consider the motion control from a candidate to next candidate for a special model. Therefore, safe navigation should be considered additionally for these methods with some special robot models. Second, the proposed algorithm is relatively simple and computationally efficient. Therefore, the computational load of the proposed randomized algorithm is smaller than the most of frontier-based algorithms such as, e.g., refs. [2, 5, 29]. Moreover, the proposed algorithm is based on a 2D range finder sensor. Comparing with other types of sensors (see, e.g., refs. [3, 30]), the range finder measurements are simpler to process and the map is built more accurately with a range finder sensor; see, e.g., refs. [3, 30]. Furthermore, a more realistic non-holonomic model of the mobile robot's motion is considered whereas many other papers in refs. [2, 28, 29, 39, 41] in the area do not take into account non-holonomic motion constraints. Additionally, unlike many other publications on the topic, a mathematically rigorous theoretical analysis of the developed algorithm is given.

The rest of the paper is organized as follows. Section 2 presents system description and relevant definitions and assumptions. The randomized navigation algorithm for area search and map building with obstacle avoidance is given in Section 3. Moreover, Section 3 contains its mathematical analysis. Computer simulations of our navigation algorithm are presented in Section 4 whereas experiments in which the proposed algorithm is implemented with a real Pioneer-3DX robot are given in Section 5. Furthermore, a comparison between our algorithm and another exploration algorithm is carried out in Section 4. Finally, Section 6 presents a brief conclusion.

2. System Description and Main Assumptions

In this paper, a planar vehicle or wheeled mobile robot is modeled as a Dubins car.[11] It travels with a constant speed and is controlled by the angular velocity limited by a given constant. The model of the vehicle is as follows (see Fig. 1):

$$\begin{array}{ll} \dot{x}(t) = v\cos\theta(t) & x(0) = x_0 \\ \dot{y}(t) = v\sin\theta(t) \quad , & y(0) = y_0 \\ \dot{\theta}(t) = u(t) \in [-u_M, u_M] & \theta(0) = \theta_0. \end{array} \tag{1}$$

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

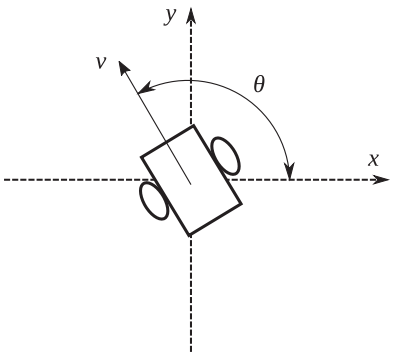


Fig. 1. Unicycle model of a wheeled mobile robot.

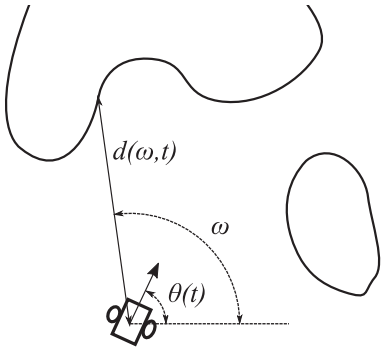


Fig. 2. Measurements of a range finder sensor.

Here, (x, y) is the vector of the vehicle's Cartesian coordinates, θ gives its orientation, v and u are the speed and angular velocity, respectively. The maximal angular velocity u_M is given. The robot satisfies the standard non-holonomic constraint

$$|u(t)| \leq u_M. \tag{2}$$

This obviously implies that the minimum turning radius of the robot is

$$R_{min} = \frac{v}{u_M}. \tag{3}$$

The non-holonomic model (1) is commonly used to describe planar motion of ground robots, unmanned aerial vehicles, and missiles; see, e.g., refs. [22, 23, 34, 35].

The wheeled robot is moving in a bounded planar unknown area $\mathcal{A}$ with several disjoint obstacles $D_1, \ldots, D_k$. Let the safety margin $d_0 > 0$ be given. The requirement is to drive the autonomous robot through the obstacle-free part of the area $\mathcal{A}$ while keeping the safety margin d_0 from both the obstacles and the boundary of area $\mathcal{A}$.

Available Measurements: The robot is equipped with a range finder sensor that at any time $t \geq 0$ measures the distance to the nearest object in all directions as follows: For any angle $0 \leq \omega < 2\pi$, the robot measures the distance $d(\omega, t)$ to the nearest boundary of an obstacle or the area $\mathcal{A}$ in the direction given by ω (see Fig. 2). Furthermore, the robot is equipped with an odometry type sensor which measures the robot's position and heading relative to its starting location and heading.

The goal of the robot is to build a complete map of the unknown cluttered area $\mathcal{A}$ while keeping a safe distance from the obstacles and the area boundary.

Definition 2.1. *Let $\mathcal{M}$ be a planar set and $\mathcal{F}$ be a binary map $\mathcal{M} \to \{0, 1\}$. The pair $(\mathcal{M}, F)$ is said to be the complete map of the area $\mathcal{A}$ if $\mathcal{M}$ and $\mathcal{A}$ are congruent, i.e. one can be transformed into the other by an isometry, and for any point $p \in \mathcal{M}$, $\mathcal{F}(p) = 1$ if and only if p under such*

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

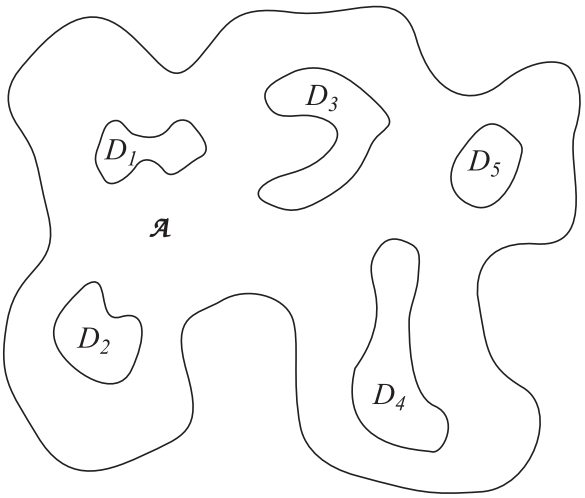


Fig. 3. An example of a complete map.

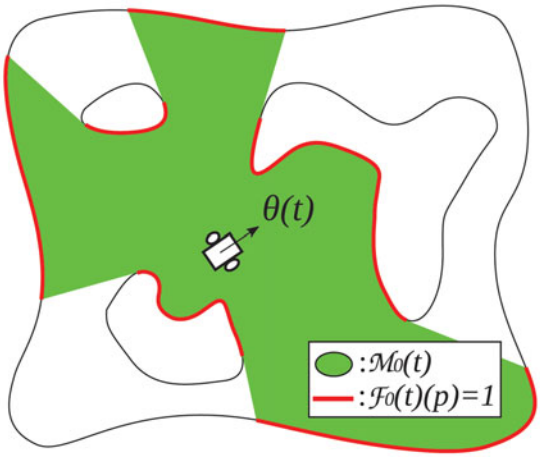


Fig. 4. Planar set $\mathcal{M}_0(t)$ and binary map $\mathcal{F}_0(t)$.

isometry corresponds to a point of either the boundary of $\mathcal{A}$ *or the boundary of one of the obstacles* $D_1, \ldots, D_k$.

In other words, a complete map means that the geometry of the area $\mathcal{A}$ is precisely known with all the obstacles $D_1, \ldots, D_k$ that belong to $\mathcal{A}$; see, e.g., Fig. 3. A topological (binary) map is used to represent the environment. This map indicates the parts of the environment where the robot cannot go and the parts where it can go. The motivation to use such a map is to get a simplest representation of the environment to make our algorithm as computationally efficient as possible.

For any robot's trajectory, the following map building algorithm is proposed.

M1: At any time $t \geq 0$, the robot builds a planar set $\mathcal{M}_0(t)$ and a binary map $\mathcal{F}_0(t)$ as follows: The set $\mathcal{M}_0(t)$ consists of all points p of the plane that are visible by the range finder sensor at time t and $\mathcal{F}_0(t)(p) = 1$ if p belongs to either obstacle boundary or area boundary, and $\mathcal{F}_0(t)(p) = 0$ otherwise; see Fig. 4. The pair $(\mathcal{M}_0(t), \mathcal{F}_0(t))$ is called the instant map at time t.

M2: At any time $t \geq 0$, the robot builds a planar set $\mathcal{M}(t)$ and a binary map $\mathcal{F}(t)$ by fusing with the help of odometry sensor all the instant maps $(\mathcal{M}_0(\tau), \mathcal{F}_0(\tau))$ for $\tau \in [0, t]$. The pair $(\mathcal{M}(t), \mathcal{F}(t))$ is called the total map at time t.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

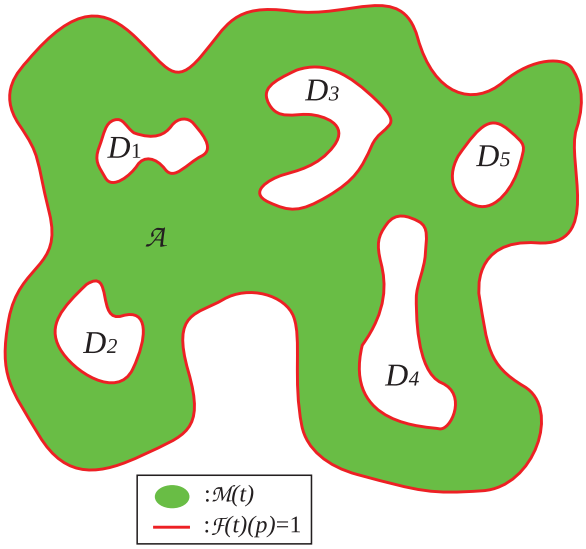


Fig. 5. An example of $\mathcal{M}(t_f)$ and $\mathcal{F}(t_f)$ which satisfy the Definition 2.2.

Now some definitions are given as follows.

Definition 2.2. *Consider a robot trajectory and the map building algorithm* **M1–M2**. *A time* $t_f \geq 0$ *is said to be a map building completing time if the total map* $\left(\mathcal{M}(t_f), \mathcal{F}(t_f)\right)$ *at time* t_f *satisfies the following property: the boundary of the set* $\mathcal{M}(t_f)$ *consists of points* p *such that* $\mathcal{F}(t_f)(p) = 1$ *and does not include any point* p *such that* $\mathcal{F}(t_f)(p) = 0$*; see Fig 5.*

Remark 2.1. If t_f is a map building completing time, it can be seen that the total map $\left(\mathcal{M}(t_f), \mathcal{F}(t_f)\right)$ at the time t_f is the complete map of the area $\mathcal{A}$. Moreover, if the robot has built by the time t_f the complete map of the area $\mathcal{A}$, then each point of the area outside of the obstacles was seen by the robot's sensor at some time.

Definition 2.3. *A trajectory* $p(t) = (x(t), y(t)), t \in [0, t_f]$ *of the robot (1) is said to be* complete map building with obstacle avoidance *if* t_f *is a map completing time and the distance between the robot's position* $p(t)$ *and the boundary of the area* $\mathcal{A}$ *and the obstacles is no less than* d_0 *for all* $t \in [0, t_f]$.

Notation 2.1. *Let* D *be any closed set,* p *be a point in the plane. Introduce the distance* $\rho(D, p)$ *as*

$$\rho(D, p) := \min_{q \in D} \|p - q\|.$$

Notice that min *is achieved since* D *is closed. Also,* $\rho(D, p) = 0$ *if* $p \in D$.

Definition 2.4. *For* $d_0 > 0$*, the* d_0-enlargement *of the domain* $D \subset \mathbf{R}^2$ *is the set* $\mathcal{E}[D, d_0]$ *formed by all points at the distance* $\leq d_0$ *from* D*, i.e.,* $\mathcal{E}[D, d_0] := \left\{p \in \mathbf{R}^2 : \rho(D, p) \leq d_0\right\}$. *On the other hand, the* d_0-reduction *of the domain* $D \subset \mathbf{R}^2$ *is the set* $\mathcal{R}[D, d_0]$ *formed by all points of* D *at the distance* $\geq d_0$ *from the boundary* ∂D *of* D*, i.e.,* $\mathcal{R}[D, d_0] := \{p \in D : \rho(\partial D, p) \geq d_0\}$.

Assumption 2.1. *The planar sets* $\mathcal{R}[\mathcal{A}, d_0]$ *and* $\mathcal{E}[D_i, d_0]$, $i = 1, \ldots, k$ *are closed, bounded, connected, and linearly connected sets.*

Assumption 2.2. *The sets* $\mathcal{E}[D_i, d_0]$ *and* $\mathcal{E}[D_j, d_0]$ *do not overlap for any* $i \neq j$. *Any set* $\mathcal{E}[D_i, d_0]$ *is a subset of* $\mathcal{R}[\mathcal{A}, d_0]$.

Remark 2.2. If Assumption 2.2 does not hold, it can be seen that map building with obstacle avoidance may be impossible (see, e.g., Fig. 8).

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

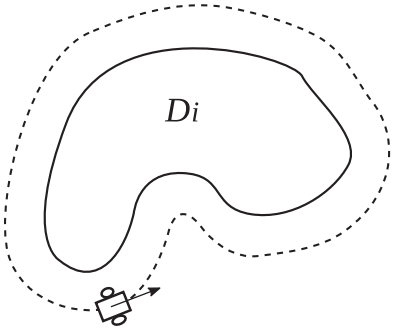


Fig. 6. The robot can see any point of the boundary of $\mathcal{A}$ or D_i.

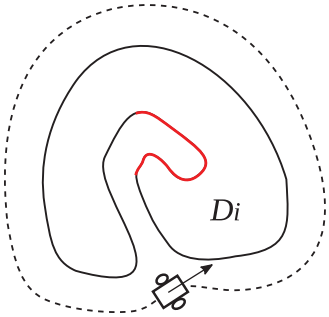


Fig. 7. There is a segment on the boundary of D_i which cannot be seen by the robot.

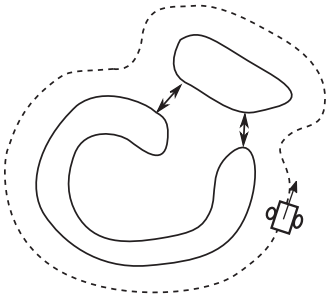


Fig. 8. The robot cannot reach and map the area which is isolated by overlapped $\mathcal{E}[D_i, d_0]$.

Assumption 2.3. *Any point of the boundaries of $\mathcal{A}$ or D_i can be connected by a straight line segment non-crossing any obstacle of the region's boundary with a point of the boundary of $\mathcal{R}[\mathcal{A}, d_0]$ or $\mathcal{E}[D_i, d_0]$, correspondingly.*

Remark 2.3. Assumption 2.3 means that any point of the boundaries of $\mathcal{A}$ or D_i can be seen by the robot travelling along the boundaries of $\mathcal{R}[\mathcal{A}, d_0]$ or $\mathcal{E}[D_i, d_0]$, correspondingly, see, e.g., Fig. 6. If Assumption 2.3 does not hold, map building with obstacle avoidance may be impossible since the boundaries of the obstacles or the region can contain some points which are impossible to see from a safe distance (see, e.g., Fig. 7).

Notation 2.2. *Let $\partial D_i(d_0)$ denote the boundary of the set $\mathcal{E}[D_i, d_0]$, and $\partial\mathcal{A}(d_0)$ denote the boundary of the set $\mathcal{R}[\mathcal{A}, d_0]$.*

Assumption 2.4. *For all i, the boundary $\partial D_i(d_0)$ is a non-self-intersecting, closed smooth curve with curvature $k_i(p)$ at any point p satisfying $|k_i(p)| \leq \frac{1}{R_{min}}$. Moreover, the boundary $\partial\mathcal{A}(d_0)$ is also a closed, non-self-intersecting smooth curve with curvature $k(p)$ at any point p satisfying $|k(p)| \leq \frac{1}{R_{min}}$.*

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

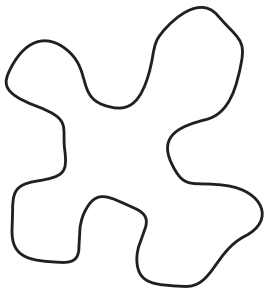


Fig. 9. There may be negative curvature at some points on $\mathcal{R}[\mathcal{A}, d_0]$ and $\mathcal{E}[D_i, d_0]$.

Remark 2.4. Notice that the sets $\mathcal{R}[\mathcal{A}, d_0]$ and $\mathcal{E}[D_i, d_0]$ are not assumed to be convex. Therefore, their boundaries may have negative curvature at some points; see, e.g., Fig. 9. The standard definition of curvature from differential geometry is used here (see, e.g., ref. [19]).

Definition 2.5. *There are two circles with the radius R_{min} that cross the initial robot position $p(0)$ and tangent to the robot initial heading $\theta(0)$. The two circles are called initial circles.*

These initial circles are assumed to be far enough from the obstacles and the boundary of $\mathcal{A}$.

Assumption 2.5. *Both the initial circles lie in the set $\mathcal{R}[\mathcal{A}, d_0]$ and do not intersect the sets $\mathcal{E}[D_i, d_0]$.*

3. Safe Navigation Algorithm

In this section, the proposed algorithm of safe navigation with map building is described. First, a number of definitions are introduced.

Definition 3.1. *A straight line L is said to be a tangent line if one of the following conditions holds:*

(1) *The line L is simultaneously tangent to two boundaries $\partial D_i(d_0)$ and $\partial D_j(d_0)$ where $i \neq j$.*
(2) *The line L is simultaneously tangent to the boundary $\partial D_i(d_0)$ and an initial circle.*
(3) *The line L is simultaneously tangent to the boundary $\partial D_i(d_0)$ and the boundary $\partial \mathcal{A}(d_0)$.*
(4) *The line L is simultaneously tangent to the boundary $\partial \mathcal{A}(d_0)$ and an initial circle.*

Points of the boundaries $\partial \mathcal{A}(d_0)$, $\partial D_i(d_0)$, and the initial circles belonging to the tangent lines are called tangent points.

For the case of simplicity, the following assumptions are introduced.

Assumption 3.1. *Any tangent point belongs to only one tangent line.*

Assumption 3.2. *If the set $\mathcal{A}(d_0)$ is non-convex, then there exists a tangent point belonging to $\partial \mathcal{A}(d_0)$.*

Remark 3.1. If the set $\mathcal{A}(d_0)$ is convex, then it can be seen that the boundary $\partial \mathcal{A}(d_0)$ does not contain tangent points. If the set $\mathcal{A}(d_0)$ is non-convex, the boundary $\partial \mathcal{A}(d_0)$ may have tangent points, see Fig. 10.

Definition 3.2. *Segments of tangent lines between two tangent points are called tangent segments. For any tangent point P, the direction of the corresponding tangent segment from this tangent point to the another tangent point of this segment is called the exit direction of the tangent point P; see Fig. 11.*

Definition 3.3. *The graph $\mathcal{G}$ is introduced that its vertices are the robot's initial position $p(0)$ and the tangent points, and its edges are the tangent segments, arcs of the initial circles and the segments of the boundaries of the extended obstacles and the reduced area that connect the vertices of the graph. The graph $\mathcal{G}$ is called the extreme graph (see, e.g., Fig. 12). A path on this graph connecting*

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

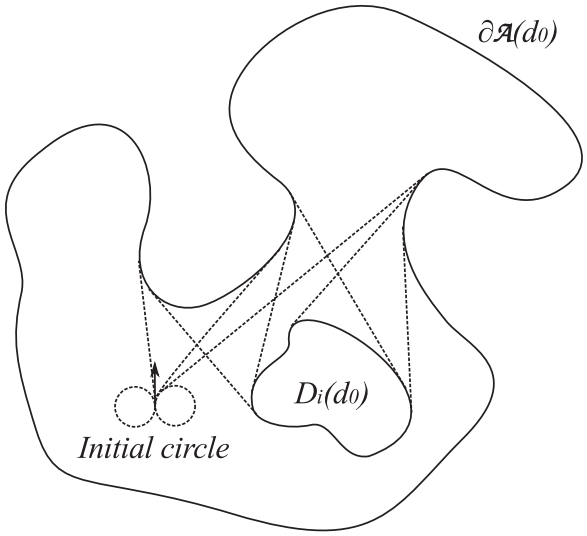


Fig. 10. Tangent lines connecting the boundary $\partial\mathcal{A}(d_0)$ with either a boundary $\partial D_i(d_0)$ or an initial circle.

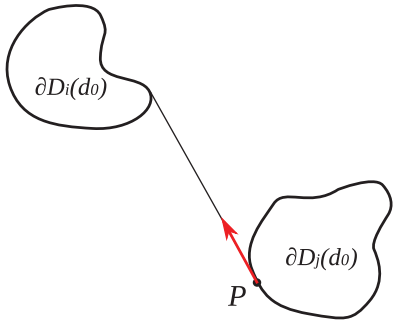


Fig. 11. The exit direction of the tangent point P.

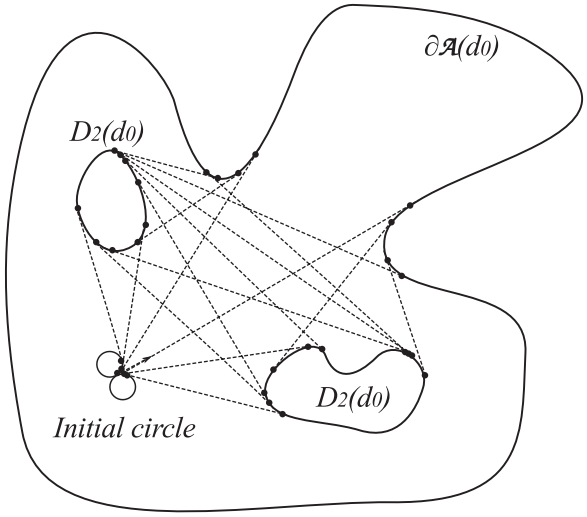


Fig. 12. The graph $\mathcal{G}$.

two vertices is said to be viable if the heading at the end of each edge of the path is equal to the heading at the beginning of the next edge of the path (see Fig. 13).

Assumption 3.3. *Let P_1 and P_2 be any two tangent points from any boundaries $\partial D_i(d_0)$ or $\partial\mathcal{A}(d_0)$. Then there exists a viable path from P_1 to P_2 on the extreme graph $\mathcal{G}$.*

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

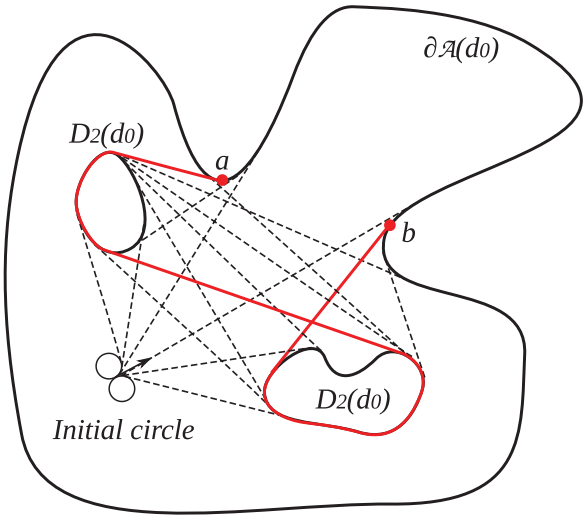


Fig. 13. A viable path connecting the vertices a and b in the graph $\mathcal{G}$.

Definition 3.4. *A tangent point on the boundary $\partial D_i(d_0)$ or $\partial\mathcal{A}(d_0)$ is said to be non-circular if the other tangent point on the corresponding tangent segment does not belong to the initial circles.*

Let $0 < q_0 < 1$ be a given number. To search and map the closed area $\mathcal{A}$ completely, the following probabilistic navigation algorithm is proposed:

A1: The robot starts to move along any of two initial circles.
A2: When the robot moving along an initial circle reaches a tangent point with the exit direction coinciding with the robot's current heading, it starts to move along the corresponding tangent segment.
A3: When the robot moves along the boundary $\partial D_i(d_0)$ or $\partial\mathcal{A}(d_0)$ and reaches a non-circular tangent point with the exit direction coinciding with the robot's current heading, with probability q_0 it starts to move along the corresponding tangent segment, and with probability $(1 - q_0)$ it continues to move along the boundary.
A4: When the robot moves along a tangent segment and reaches a tangent point on the boundary $\partial D_i(d_0)$ or $\partial\mathcal{A}(d_0)$, it starts to move along this boundary.

The navigation algorithm **A1–A4** is based on switching between following the initial circles, tangent lines, and segments of boundaries of the extended obstacles or the reduced search area. In the step **A1**, the proposed algorithm makes the robot either move along the corresponding tangent segment (with probability q_0) or continue to move along the boundary (with probability $(1 - q_0)$. Hence, the algorithm **A1–A4** employs a degree of randomness as part of its logic and therefore belongs to the class of randomized algorithms.

Now the main theoretical result of this paper is presented as follows.

Theorem 3.1. *Suppose that Assumptions* 2.1–2.5, 3.1–3.3 *hold and the robot is navigated by the algorithm* **A1–A4** *with some $q_0 \in (0, 1)$. Then for any initial position and heading of the robot, with probability* 1 *there exists a time $t_f \geq 0$ such that the trajectory $p(t) = (x(t), y(t)), t \in [0, t_f]$ of the robot (1) is complete map building with obstacle avoidance.*

Assumptions 2.1–2.5, 3.1–3.3 are needed for mathematically rigorous proof of the main theoretical result (Theorem 3.1). In practice, the proposed algorithm often performs well even when some of them do not hold.

The proof of Theorem 3.1 The first step in the proof of Theorem 3.1 is to prove that with probability 1 there exists a time $t_f \geq 0$ which is a map building completing time (see Definition 2.2). Indeed, there are the following two possible cases:

Case 1: The set $\mathcal{R}[\mathcal{A}, d_0]$ is non-convex.
Case 2: The set $\mathcal{R}[\mathcal{A}, d_0]$ is convex.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

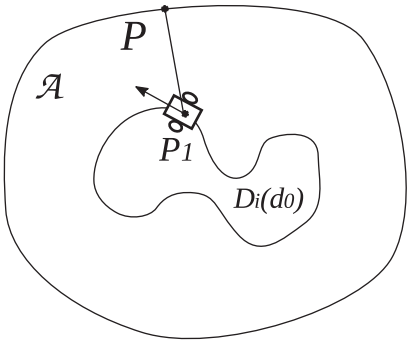


Fig. 14. A straight line segment connecting P with a point P_1 of a boundary $\partial D_i(d_0)$.

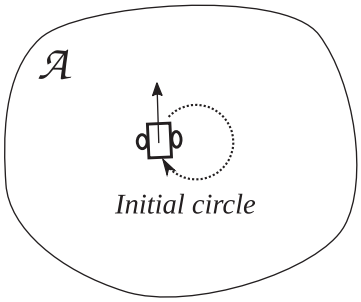


Fig. 15. The map is built after a complete turn of an initial circle.

In **Case 1**, Assumption 3.2 implies that the algorithm **A1–A4** includes switching between tangent lines and the boundaries $\partial\mathcal{A}(d_0)$ and $\partial D_i(d_0)$ with some non-zero probabilities. Therefore, it follows from Assumption 3.3 that with probability 1 there exists a time $t_f \geq 0$ such that by this time the robot's trajectory includes all the points of the boundaries $\partial\mathcal{A}(d_0)$ and $\partial D_i(d_0)$ for all i (notice that the robot can travel along the boundaries in either clockwise or counter-clockwise direction). Now it immediately follows from Assumption 2.3 that the robot by the time t_f has seen every point of the boundaries of the region and the obstacles. Furthermore, the navigation algorithm **A1–A4** consists of moving along the boundaries $\partial D_i(d_0)$, $\partial\mathcal{A}(d_0)$, two initial circles, and tangent lines between them. According to Assumption 2.5, the distance between both initial circles and obstacles and the boundary of the region is greater or equals d_0. Therefore, under the navigation algorithm **A1–A4**, the robot always moves along lines such that the distance between these lines and the obstacles and the boundary of the region is greater or equals d_0. Hence, the distance between the robot and the obstacles and the boundary of the region is greater or equals d_0.

In **Case 2**, there is no a tangent line connecting a tangent point on the boundary $\partial\mathcal{A}(d_0)$ with either a boundary $\partial D_i(d_0)$ or an initial circle since $\partial\mathcal{A}(d_0)$ does not have tangent points. In this case, the robot does not travel along $\partial\mathcal{A}(d_0)$. The algorithm **A1–A4** includes switching between tangent lines and the boundaries $\partial D_i(d_0)$ with some non-zero probabilities. Therefore, it follows from Assumption 3.3 that with probability 1 there exists a time $t_f \geq 0$ such that by this time the robot's trajectory includes all the points of the boundaries $\partial D_i(d_0)$ for all i. Assumption 2.3 immediately implies that the robot by the time t_f has seen every point of the boundaries of the obstacles. Moreover, this and the convexity of the set $\mathcal{R}[\mathcal{A}, d_0]$ imply that the robot by the time t_f has seen every point of the boundary of the region $\mathcal{A}$ as well. Indeed, Let P be an arbitrary point of the boundary of $\mathcal{A}$. If there exists a straight line segment connecting P with a point P_1 of a boundary $\partial D_i(d_0)$ which does not cross any obstacle (see Fig. 14) than the robot sensed the point P when it was at P_1. If there is no such a straight line segment then the region $\mathcal{A}$ does not contain any obstacles. In this case, the algorithm **A1–A4** results in travelling along an initial circle and the map of the region will be built after one complete turn (see Fig. 15). The proof of the statement that the algorithm **A1–A4** guarantees that the distance between the robot and the obstacles and the boundary of the region is greater or equals d_0 is exactly the same as in **Case 1**. This completes the proof of Theorem 3.1.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

Remark 3.2. It should be pointed out that the robot will stop at a time $t_f \geq 0$ that is a map building completing time (see Definition 2.2). It immediately follows from Remark 2.1 and Theorem 3.1 that trajectory $p(t) = (x(t), y(t)), t \in [0, t_f]$ of the robot (1) navigated by the algorithm **A1–A4** is complete map building with obstacle avoidance.

4. Computer Simulations

In this section, computer simulations of the collision free map building navigation algorithm **A1–A4** are presented. In these simulations, the odometry errors, range finder noise, and dynamic environments are discussed. Finally, our algorithm is compared with another environment exploration and map building algorithm.

To follow tangent lines and boundaries of the reduced region and the extended obstacles, a sliding mode control law which is a modification of the law of ref. [34] is used, that consists of switching between a boundary following approach proposed in ref. [24], and the pure pursuit navigation approach; see, e.g., ref [36]. Our sliding mode navigation law can be expressed as follows:

$$u(t) = \begin{cases} \pm u_M & R1 \\ \Gamma sgn\left[\phi_{tan}(t)\right] u_M & R2, \\ \Gamma sgn\left[\dot{d}_{min}(t) + X(d_{min}(t) - d_0)\right] u_M & R3 \end{cases} \tag{4}$$

where the function $sgn(x)$ is defined as follows:

$$sgn(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0. \\ -1 & x < 0 \end{cases} \tag{5}$$

This navigation law defined three control strategies corresponding to three separate modes $R1 - R3$. There are three rules for switching the mode between three modes $R1 - R3$. Initially, mode $R1$ is active, and transitions to other modes are determined as follows:

(1) $R1 \rightarrow R2$: It occurs when the robot is at a tangent point on one of the initial circles and the exit direction at the tangent point coincides with the robot's current heading.
(2) $R2 \rightarrow R3$: It occurs when the robot is at a tangent point on a tangent segment and reaches the boundary $\partial D_i(d_0)$ or $\partial \mathcal{A}(d_0)$.
(3) $R3 \rightarrow R2$: With the probability q_0, it occurs when the robot is at a tangent point on the boundary $\partial D_i(d_0)$ or $\partial \mathcal{A}(d_0)$ and the exit direction at the tangent point coincides with the robot's current heading.

Mode $R1$ describes motion along the initial circle with maximal angular velocity. Mode $R2$ describes pursuit navigation, where $\phi_{tan}(t)$ is defined as the angle between the robot's heading and a line segment connecting the vehicle and the currently tracked tangent segment. Mode $R3$ describes boundary following navigation, where the control calculation is based on the minimum distance to the nearest obstacle, defined as $d_{min}(t)$.

This control law is subject to some restrictions which are inherited from ref. [24]. The variable Γ is defined as $+1$ if the boundary followed is on the left of the tangent being tracked, -1 if it is on the right. A constant $d_{trig} > d_0$ is also introduced to determine when the control system transitions to boundary following mode.[24] The saturation function X is defined as follows:

$$X(r) = \begin{cases} lr & |r| < k \\ lk\ sgn(r) & otherwise, \end{cases} \tag{6}$$

where l and k are tunable constants. Because of any potential chattering in the robot's heading caused by the sliding mode control, once it is decided to not pursue a tangent line, there is a short pause until tangent following can potentially be engaged again. It means, during this short pause, the mode $R3$ will not switch to $R2$ even if the robot's heading coincides with a tangent line again.

According to the system description in Section 2, the robot is equipped with a range finder sensor and an odometry sensor. The range finder sensor can inform the distance from the mobile robot to the

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

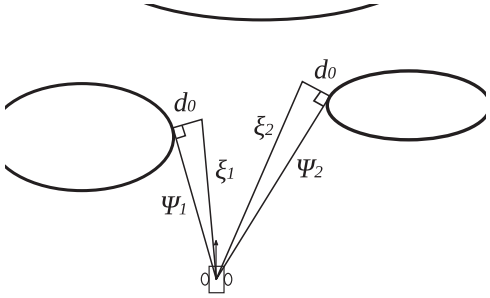


Fig. 16. The calculation of the tangent line ξ_k.

nearest obstacles in a finite number of directions. The measurements of the range finder sensor can be represented as a finite point set $\{W_i(\omega_i, \varsigma_i)|i = 1, 2 \ldots n\}$. ς_i is the distance from the robot to the nearest obstacle in the direction ω_i. ω_i is the direction angle relative to the robot's heading. Moreover, the odometry sensor measures the robot's speed and angular velocity. The robot can calculate its position and orientation by using odometry.

Notice that the robot will stop at a time $t_f \geq 0$ that is a map building completing time (see Definition 2.2). As it is pointed out in Remark 3.2, in this case, the trajectory $p(t) = (x(t), y(t)), t \in [0, t_f]$ of the robot (1) navigated by the algorithm **A1–A4** is complete map building with obstacle avoidance.

Now, the following approaches to implement the proposed algorithm **A1–A4** are presented.

With a small angular resolution $\Delta\omega$ of the range finder sensor, the variable $d_{min}(t)$ can be approximated by the minimum value ς_{min}. Similarly, $\dot{d}_{min}(t)$ can be approximated by $\frac{d_{min}(t+T)-d_{min}(t)}{T}$ with a short sampling interval T.

To determine when the mode $R2$ should be activated, the following steps are designed to check whether the robot's orientation coincides with a tangent line (see Fig. 16):

(1) Calculate $|\Delta\varsigma_i|$ between successive ς_i and ς_{i+1} for any $i = 1, 2 \ldots n-1$. According to the Assumption 2.2, the minimum distance between any two obstacles is $> 2 \cdot d_0$. Therefore, the tangent lines on the boundary of obstacles are obtained by checking if $|\Delta\varsigma_i| > 2 \cdot d_0$. When there exists any $|\Delta\varsigma_k| > 2 \cdot d_0$, let ς'_k denote the smaller value between ς_k and ς_{k+1}. Then, the line segment Ψ_k from the pole to the point $(\omega'_k, \varsigma'_k)$ is a tangent line on an obstacle.
(2) According to the geometry, let Ψ_k be a leg and construct a right triangle. The length of another leg which shares the common endpoint $(\omega'_k, \varsigma'_k)$ with Ψ_k is d_0. After this, the hypotenuse ξ_k can approximately represent the tangent line on the corresponding safe boundary $\partial D(d_0)$. Note that if $\varsigma_k < \rho_{k+1}$, the polar angle of ξ_k is bigger than the polar angle of Ψ_k (see ξ_2 in Fig. 16). If $\varsigma_k > \rho_{k+1}$, the situation is opposite (see ξ_1 in Fig. 16).
(3) Define a safe area Ω_k for each line segment ξ_k (see Fig. 17). The set Ω_k is formed by all points at the distance $< d_0$ from the line segment ξ_k. If there is any point W_i within the area Ω_k, ξ_k is not a real tangent line because in geometry, it intersects with other $\partial D(d_0)$ and can not be pursued safely by the robot.
(4) If there exist a ξ_k that the absolute value of its angle ω_k in the local polar coordinate system is smaller than a given threshold θ_{trig}, the robot's orientation approximately coincides with the tangent line ξ_k.

To calculate $\phi_{tan}(t)$ in the Eq. (4), an intermediate target P_T is defined. Assume that when the mode $R2$ is activated, the robot's orientation coincides with the tangent line ξ_k. Then, the temporary target P_T is the endpoint of ξ_k. After this, $\phi_{tan}(t)$ is the angle between the robot's heading and the vector from the robot's position $(x(t), y(t))$ to the intermediate target P_T.

A raster graphics map is used to represent the environment. Each cell of the map has three optional states (unknown, unoccupied, and occupied). Initially, all the cells are unknown. At each sampling time, the measurements $W_i(\omega_i, \varsigma_i)$ are transformed from the local polar coordinate system to the global Cartesian coordinate system $W_i(x_{W(i)}, y_{W(i)})$. Then, the Bresenham's line algorithm[7] is used to determine the state of each cell intersecting with the line segments between robot's

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

Table I. The Parameters for Simulation 1.

Speed of robot	v	1 m/s
Maximum angular velocity	u_M	1.4 rad/s
Minimum turning radius	R_{min}	0.71 m
Safety margin	d_0	1.7 m
Sampling interval	T	0.25 s
Angle threshold	θ_{trig}	0.08 rad
Distance threshold	d_{trig}	0.3 m
Probability	q_0	0.4
Resolution of the map		20 cm^2/cell

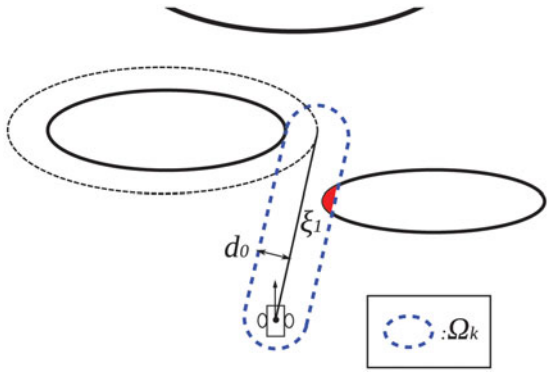


Fig. 17. The definition of safe area Ω_k.

position $(x(t), y(t))$ and each $W_i(x_{W(i)}, y_{W(i)})$. The state of each cell is updated by the following strategy:

(1) If an unknown cell or an unoccupied cell is occupied by $W_i(x_{W(i)}, y_{W(i)})$, the cell will be updated to occupied cell.
(2) If an unknown cell intersects with a line segment between $(x(t), y(t))$ and $W_i(x_{W(i)}, y_{W(i)})$ and is not occupied by $W_i(x_{W(i)}, y_{W(i)})$, the cell will be updated to unoccupied cell.

In the implementations of the proposed algorithm, there are some parameters which should be determined as follows:

(1) The maximal angular velocity u_M is a known physical parameter of a mobile robot.
(2) For a real robot, the safety margin d_0 should be bigger than the maximum distance from the robot's center to its boundary. Moreover, it should be smaller than a half of the minimal distance between any obstacles (see Assumption 2.2).
(3) The robot's speed v should be $< u_M \cdot d_0$ because R_{min} should be $< d_0$ (see Assumption 2.4).
(4) The two thresholds θ_{trig} and d_{trig} should be tuned in different implementations to guarantee the correct mode transitions.
(5) If the precision of a range finder sensor is $\pm e$, the resolution of the map is recommended to be $> (2e)^2/cell$ because a higher resolution than $(2e)^2/cell$ is unnecessary.

In the following parts, five groups of computer simulations are carried out to confirm and discuss the performance of the proposed algorithm.

Simulation 1: The proposed algorithm was simulated in two 112.5 m × 112.5 m cluttered environments with different types of obstacles. In the first scene, there are some non-convex obstacles. The parameters used for the simulation are listed in Table. I. The result of the simulation in the first scene is shown in Fig. 18. It can be seen that the robot searched the whole area successfully and built a complete map of the environment in 920.5 s.

In the second scene (a maze), the safety margin was changed to 4 m to make the result clear to see. Thus, the speed and angular velocity used for this simulation were changed to 0.8 m/s and 0.5 rad/s.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

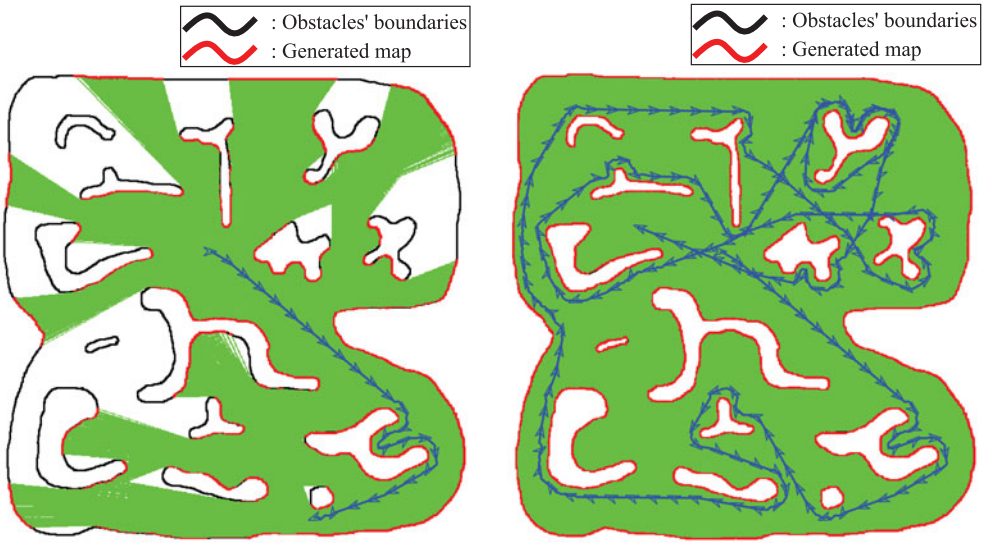


Fig. 18. The result of the simulation in the first scene of Simulation 1. (a) $t = 125$ s. (b) $t = 920.5$ s.

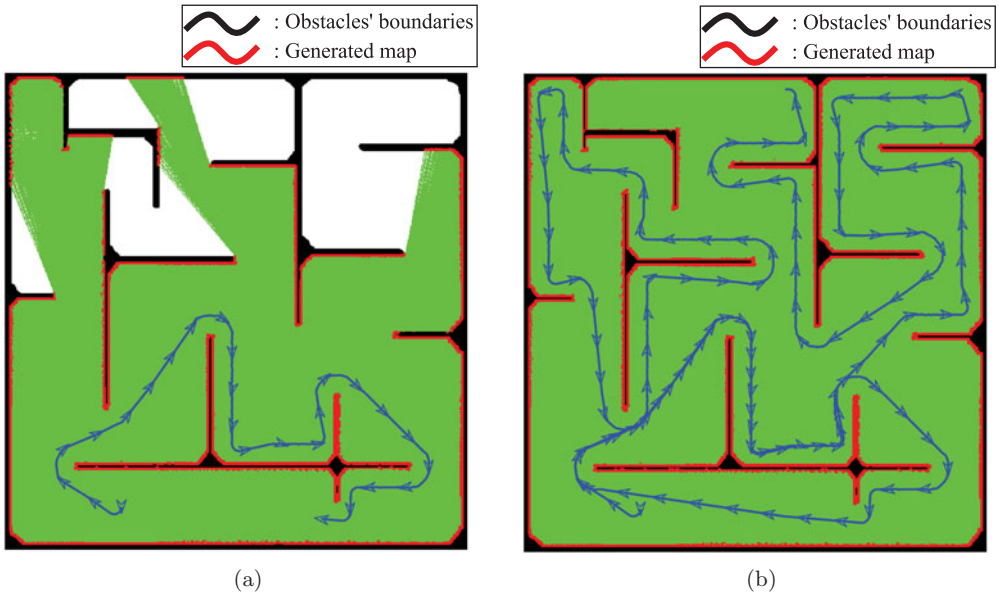


Fig. 19. The result of the simulation in the second scene of Simulation 1. (a) $t = 250$ s. (b) $t = 1186$ s.

As the result shows in Fig. 19, the robot started from an initial position and searched the whole area of this maze without any collision. Finally, it took 1168 s to build a complete map of this environment.

Simulation 2: In Simulation 2, the range finder sensor is considered as a LRF and the noise model proposed in ref. [16] was used to simulate the LRF noise. The modeled noise is a combination of flicker noise and white noise with standard deviation of 1.77023 mm and 3.03856 mm (see Fig. 20). In this simulation, the modeled noise was increased by five times to make the effects of the noise more obvious in the result of the simulation. The size of the environment is 11.25 m × 11.25 m and the resolution of the map is 1 cm^2/cell. Figure 21 shows the complete map in this simulation with the LRF noise. It can be seen that the proposed algorithm is robust with the range finder errors, although the map quality is influenced (i.e., rough boundaries in generated maps).

Simulation 3: In the proposed algorithm, odometry is used to estimate the position and orientation of the robot. The odometry errors are caused by many reasons.[6] In this simulation, only the odometry errors caused by the approximation in the odometry equations[6] are considered. The sampling interval

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

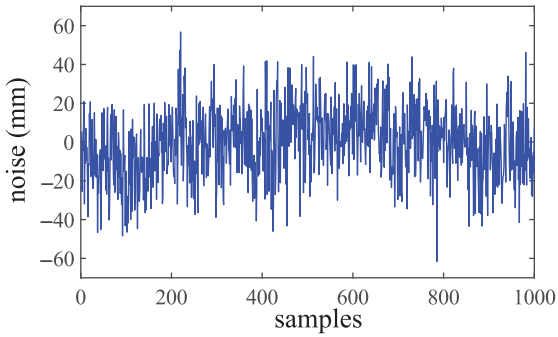


Fig. 20. The modeled laser range finder noise.

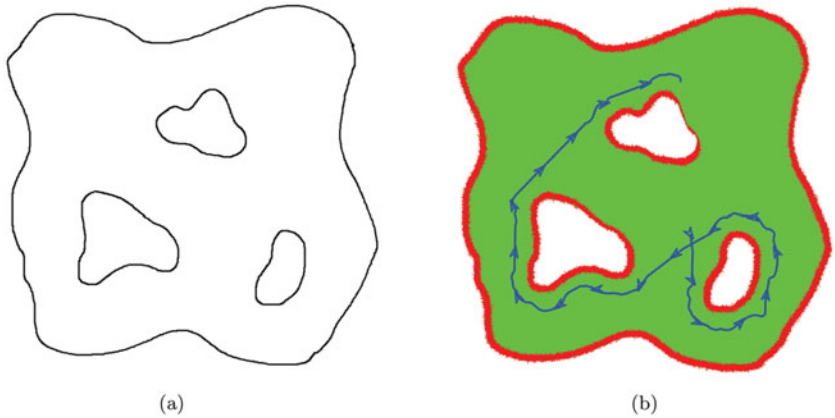


Fig. 21. The result of Simulation 2 with range finder noise. (a) Obstacles' boundaries. (b) Generated map.

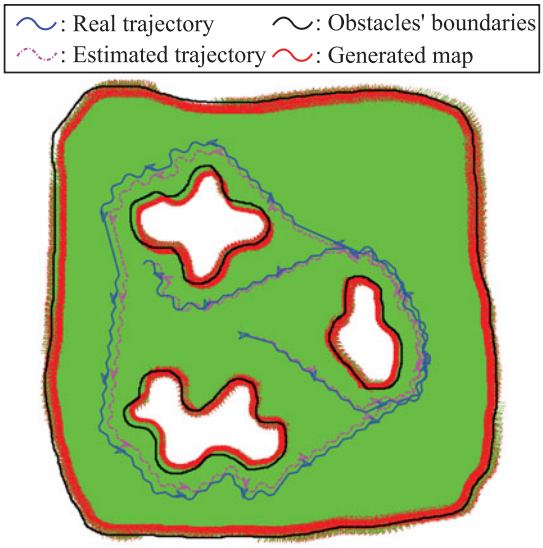


Fig. 22. The result of Simulation 3 with odometry errors.

in this simulation is 0.5 s, which caused an obvious odometry errors. The size of the environment is 11.25 m × 11.25 m. The result of the simulation (see Fig. 22) shows that the estimation of the robot's position diverged from the real position gradually with the increase of time. As a result of odometry errors, there are obvious errors on the generated map.

To alleviate the odometry errors, an improved estimation model with the help of an accelerometer is proposed here. The accelerometer measures the acceleration of the robot in the direction vertical

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

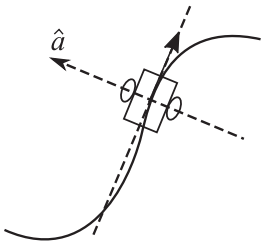


Fig. 23. The direction of the acceleration measured by the accelerometer.

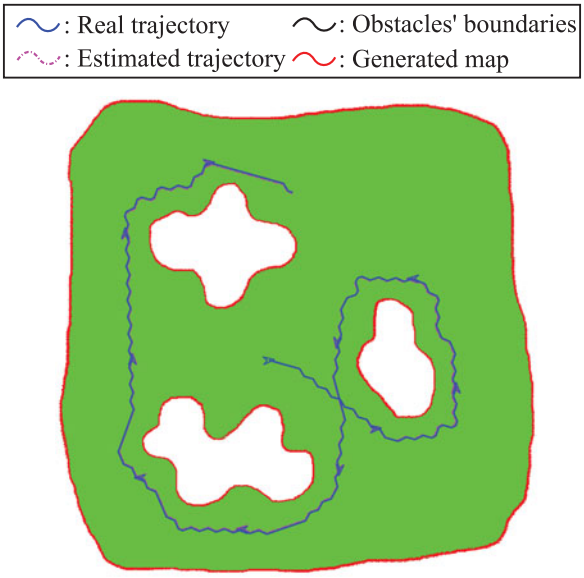


Fig. 24. The result of the simulation with the improved estimation model (7). Notice that the obstacles' boundaries nearly coincide with the generated map and the real trajectory of the robot nearly coincides with the estimated trajectory.

to the robot's heading (see Fig. 23). Let $\hat{v}$ and $\hat{u}$ be the speed and angular velocity measured by odometry sensors, respectively. Let $\hat{a}$ be the acceleration measured by the accelerometer. In the improved estimation model, $\hat{v}$, $\hat{u}$ and $\hat{a}$ are approximated as constant during the sampling interval T. Based on the approximation, the improved estimation model is proposed as follows:

$$\begin{aligned} x(k+1) &= x(k) + Q\cos\theta(k) + \tfrac{\hat{a}T^2}{2}\cos(\theta(k)+\tfrac{\pi}{2}) \\ y(k+1) &= y(k) + Q\sin\theta(k) + \tfrac{\hat{a}T^2}{2}\sin(\theta(k)+\tfrac{\pi}{2}) \\ \theta(k+1) &= \theta(k) + \hat{u}(k) \end{aligned} , \tag{7}$$

where

$$Q = \int_0^T \sqrt{\hat{v}^2 - (\hat{a}t)^2}\,\mathrm{d}t = (\frac{T}{2}\sqrt{\hat{v}^2 - (\hat{a}T)^2} + \frac{\hat{v}^2}{2\hat{a}}\arcsin(\frac{\hat{a}T}{\hat{v}^2})). \tag{8}$$

To confirm the performance of the improved estimation model (7), a new simulation is carried out with this improved model in a same environment. The result of the new simulation is shown in Fig. 24. Comparing with Fig. 22, the odometry errors are alleviated significantly by the improved estimation model (7). Additionally, for other types of odometry errors, there are many methods focusing on the alleviation of odometry errors, e.g., refs. [9, 10, 33].

Simulation 4: In Simulation 4, a dynamic environment with both steady and moving obstacles is considered. In the dynamic environment, the proposed algorithm should be modified to explore the environment safely and build the complete map correctly. In this simulation, the robot navigation algorithm contains a simple motion detector block to distinguish moving obstacles and avoid collisions

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

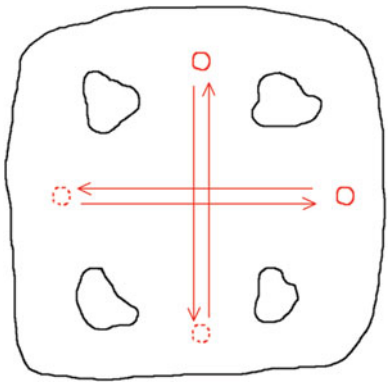


Fig. 25. The dynamic environment in Simulation 4.

with them. There are many methods focusing on the moving obstacle detection, e.g., [26,32,43]. Moreover, in our future research, more advanced methods of avoiding collisions with moving obstacles can be integrated with the proposed algorithm (see, e.g., refs. [14, 23]). With the motion detector block, the robot can avoid pursuing a tangent segment which is tangent to a moving obstacles (see algorithm **A3**). With the update of the map during the exploration, the boundaries of moving obstacles on the map will be eliminated completely and it will end up with a correct map of the steady environment (the region and all its steady obstacles).

With the above modifications, Simulation 4 is carried out with two moving obstacles in a 5 m $\times$ 5 m environment (see Fig. 25). The result of the simulation is shown in Fig. 26. It can be seen that the robot successfully explored the area without pursuing any tangent line belonging to moving obstacles. Moreover, it can be seen that, at the time 47.1 s, the robot avoided the moving obstacle when pursing a tangent line. According to the update of the map, it can be seen that the boundaries of moving obstacles are eliminated gradually. Finally, the robot built a correct complete map of the region and all its steady obstacles.

Simulation 5: In recent researches, the frontier-based exploration algorithms are the most common exploration strategies. In this simulation, a frontier-based exploration algorithm proposed in ref. [39] was selected to compare the exploration time with the proposed algorithm. The selected algorithm has a better performance than current exploration methods. It keeps the map information integrated during exploration and uses a multi-criteria decision method to overcome the drawbacks of the traditional weighted average method to improve the efficiency of the exploration.

Notice that, although the Growing Neural Gas network topological map is used to represent the environment, which is different from geometrical map, it does not affect the exploration time. In this simulation, the algorithm of ref. [39] was simulated and run in some new examples to compare the exploration time with the proposed algorithm.

In these new examples, the exploration time of the two algorithms is compared in different environment with different number of obstacles. Notice that the algorithm of ref. [39] refers to a front exploration area which is a semi-circle at the front of the robot. Therefore, different radius of the front exploration area was considered in the simulations.

In the comparison, the size of the environment is 37.5 m $\times$ 37.5 m and the robot's speed is 0.2 m/s. Figure 27 shows the different exploration time of two algorithms with different number of obstacles. The performance of the proposed algorithm refers to Line A. The performance of the algorithm of ref. [39] with different radius of 5 m, 6 m, 7 m, and 8 m refers to Line B1 to B4, respectively.

According to the comparison, it can be seen that our randomized search algorithm has a strong advantage because our method builds a complete map of the environment nearly twice faster than the algorithm of ref. [39]. Furthermore, our algorithm is designed for a non-holonomic model (see Eq. (1)), whereas the algorithm of ref. [39] and the most of works in this fields do not consider any non-holonomic constraint of the robot's motion. Moreover, the algorithm of ref. [39] may fail to search some part of the area when the diameter of the front exploration area is smaller than the minimum distance between any obstacles. Additionally, the performance of our algorithm is verified

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

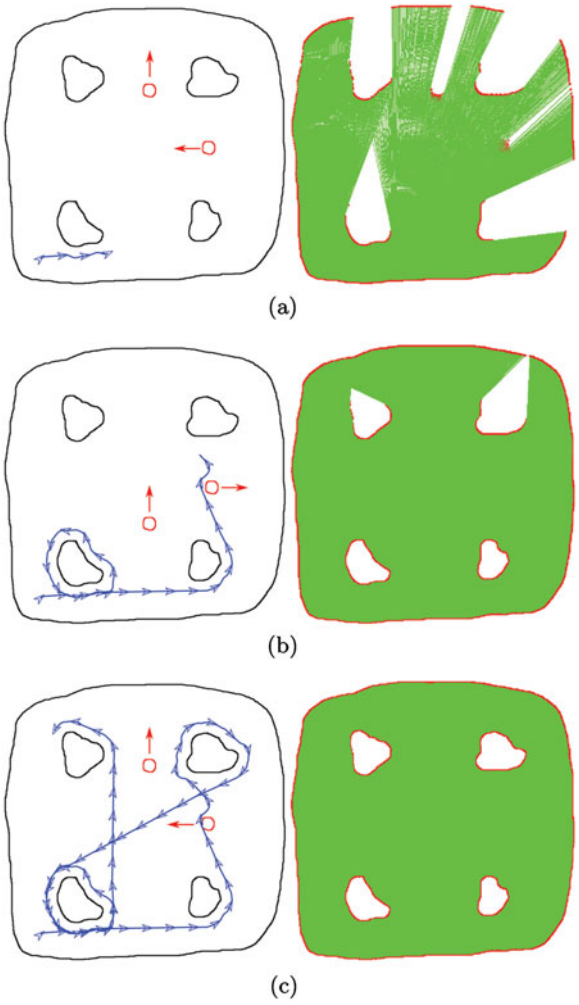


Fig. 26. The result of Simulation 4 in the dynamic environment. (a) $t = 6.1$ s. (b) $t = 47.1$ s. (c) $t = 112.2$ s.

by a mathematically rigorous proof. Finally, the developed randomized algorithm requires a smaller computational load than the exploration algorithm proposed in ref. [39].

5. Experiments with a Real Wheeled Robot

In this section, the real implementation of the proposed collision free map building algorithm **A1–A4** are proposed with a Pioneer-3DX wheeled mobile robot. The robot has an on-board computer with a Linux operating system. It can be programmed to control the speed of each wheel separately and process the data of the mounted sensors. The Pioneer-3DX robot calculates its real-time position and orientation by odometry.

In the experiments, the robot is equipped with a SICK LMS-200 LRF and an odometry sensor. This LRF's angular resolution, scanning range, and measurement range are 0.5°, 180°, and 80 m, respectively. The LRF's precision is $\pm$ 15 mm. Notice that, to guarantee that the minimum distance to obstacles can be measured correctly and any point on obstacles' boundaries can be detected by the range finder, the scanning range of the LRF should be $\geq$180°. If the scanning range is $<$180°, the minimum distance cannot be measured correctly and some parts of obstacles' boundaries may never be detected (see Fig. 28). The change of the scanning range does not affect the final result of the map building and safe navigation. However, with the decrease of the scanning range, the map building completing time will increase. Moreover, it is noticed that the scene we built in the experiments is not quite large. Thus, the odometry errors of the robot is quite small and the generated map is accurate. If the proposed algorithm is implemented in a larger scene, the approach proposed in Simulation 3

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

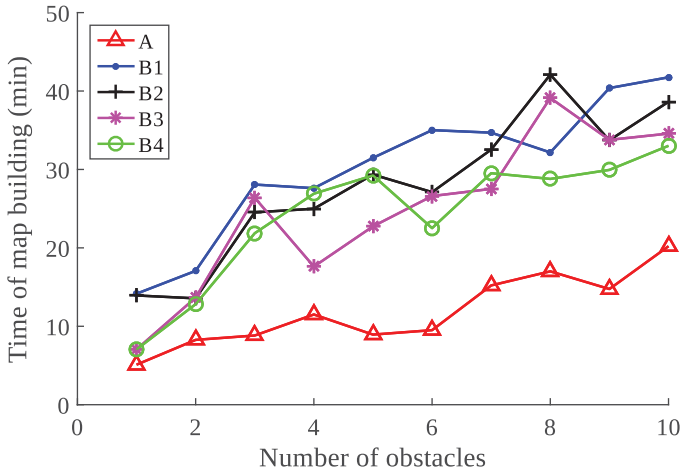


Fig. 27. The exploration time of the two algorithms with different number of obstacles.

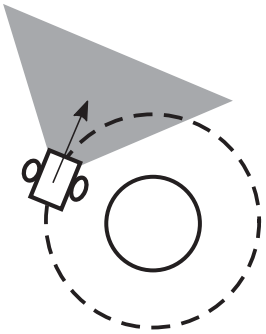


Fig. 28. The minimum distance cannot be measured correctly and obstacle's boundary cannot be detected with the scanning range which is $<180^{\circ}$.

can be used to reduce odometry errors. Other technologies can also be used to reduce the localization error of the robot such as the use of Global Positioning System (GPS) in outdoor environments.

A closed area in an indoor environment was built by blocking the hallway and doors. Seven obstacles were placed on the floor. Each obstacle was higher than the robot to guarantee that the obstacle can be detected by the LRF. Considering the measurement range of the LRF, the maximum distance from any point to any boundary should be <80 m.

The robot was programmed with the proposed algorithm and put into the environment. When the robot started running, its mode R1 was activated and it moved along the initial circle firstly (see **A1**). When its heading coincided with a tangent line ξ (See Fig. 16), its mode was transitioned to R2 from R1 or R3 (see **A2** and **A3**). When the robot reached the endpoint of the line segment ξ, its mode was transitioned back to R3 (see **A4**) and it started tracking $\partial D_i(d_0)$ or $\partial \mathcal{A}(d_0)$. During the period of the safe navigation, the robot was continuously updating the map and checked whether the map building is completed (see Definition 2.2). Finally, the robot stopped when the complete map was built.

Remark 5.1. The saturation function (6) is used to replace the sign function to reduce control chattering, which is undesired in experiments with a real robot.[36]

In the following parts, five experiments are carried out with different probability q_0 in the preset environment to confirm the performance of the proposed algorithm. The parameters used in these experiments are listed in Table II. The probability q_0 for the five experiments are 0.3, 0.4, 0.5, 0.6, and 0.7.

At the beginning, the robot selected one of two initial circles randomly to move. When the robot's heading coincided with a tangent segment, the robot left the initial circle and reached the enlarged boundary by pursuing this tangent segment. When reached the enlarged boundary, the robot started patrolling this boundary and randomly selected the next tangent segment to pursue. Finally, the

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

Table II. The parameters for experiment 1.

Speed of robot	v	0.15 m/s
Maximum angular velocity	u_M	0.4 rad/s
Minimum turning radius	R_{min}	0.375 m
Safety margin	d_0	0.55 m
Sampling interval	T	0.2 s
Angle threshold	θ_{trig}	0.1 rad
Distance threshold	d_{trig}	0.1 m
Resolution of the map		3 cm^2/cell

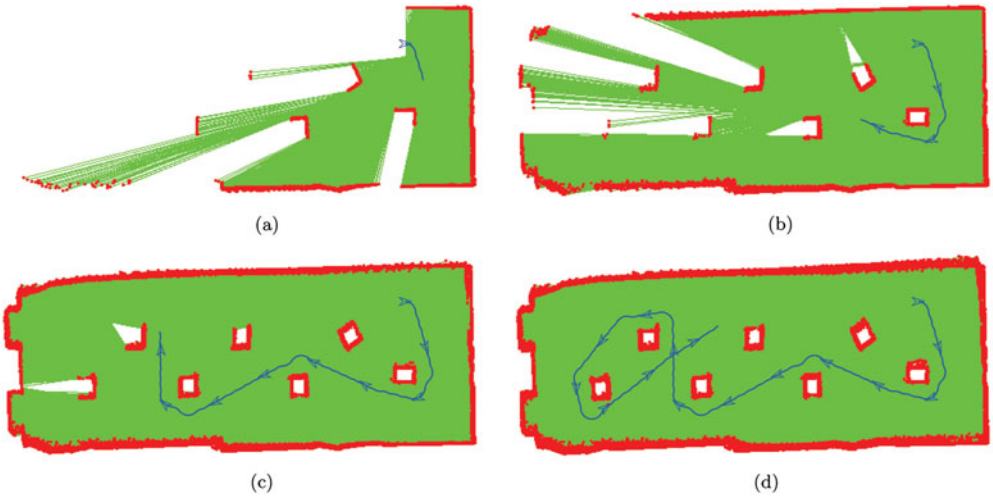


Fig. 29. The generated maps at different time with probability $q_0 = 0.6$. (a) $t = 10$ s. (b) $t = 40$ s. (c) $t = 100$ s. (d) $t = 174.2$ s.

robot explored the closed area completely and built a correct complete map of this environment. This procedure is indicated in Figs. 29 and 30. Figure 29 shows the generated maps and the robot's trajectory at different time in the experiment with probability $q_0 = 0.6$. Figure 30 shows some pictures of the corresponding experiment.

The results of the five experiments including the generated complete maps, map building completing time t_f, the robot trajectories, and corresponding probability q_0 are indicated in Fig. 31. It can be seen that, in the five experiments, the robot successfully avoided all the obstacles and built the complete maps of the closed area in a finite time. Moreover, according to the comparison of the experimental results, it can be seen that the map building completing time t_f and the length of the robot trajectory are influenced by different values of the parameter q_0. Furthermore, it is easy to see that the accuracy of the generated map is influenced by the length of the robot trajectory because a longer robot trajectory involves bigger odometry errors and reduces the accuracy of the map (e.g., Fig. 31).

6. Conclusion and Future Work

In this paper, a safe area search and map building algorithm for a mobile ground robot in a closed unknown 2D environment with obstacles was proposed. The obstacles are not required to be convex. The developed algorithm belongs to the class of randomized algorithms. It was proved that with probability 1 the robot finishes searching the area and builds a complete map of the environment in a finite time. Moreover, the odometry errors, range finder noise, and dynamic environments are considered and discussed in the computer simulations. The computer simulations and experiments with a real Pioneer-3DX robot confirmed the expected performance of the proposed navigation algorithm.

The proposed method has a number of advantages. First, the area search and safe navigation were combined together in our method. Second, the proposed algorithm is relatively simple and

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

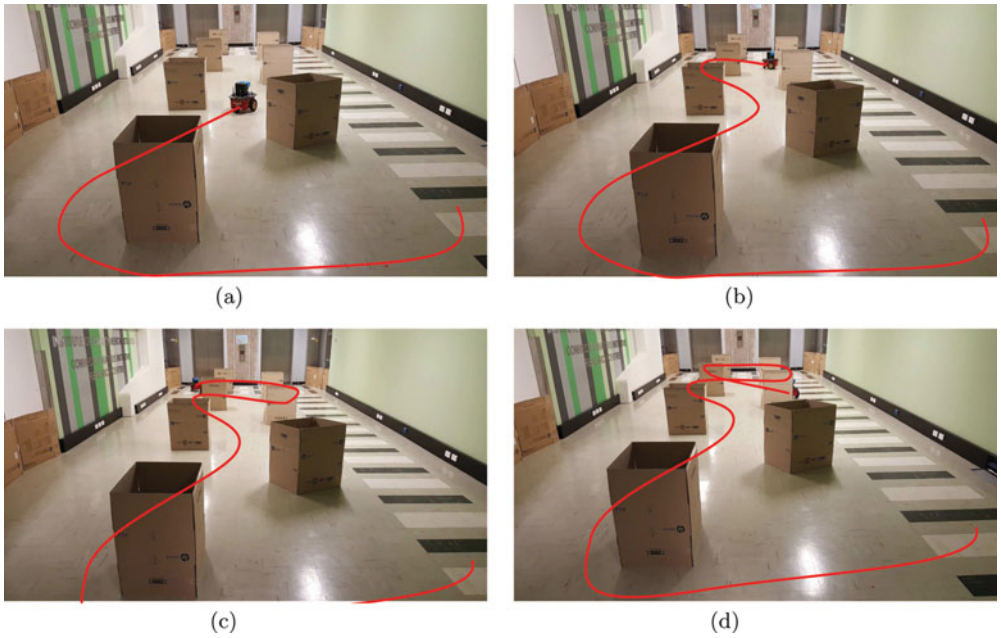


Fig. 30. The experiment pictures at different time with probability $q_0 = 0.6$. (a) $t = 51$ s. (b) $t = 97$ s. (c) $t = 143$ s. (d) $t = 174.2$ s.

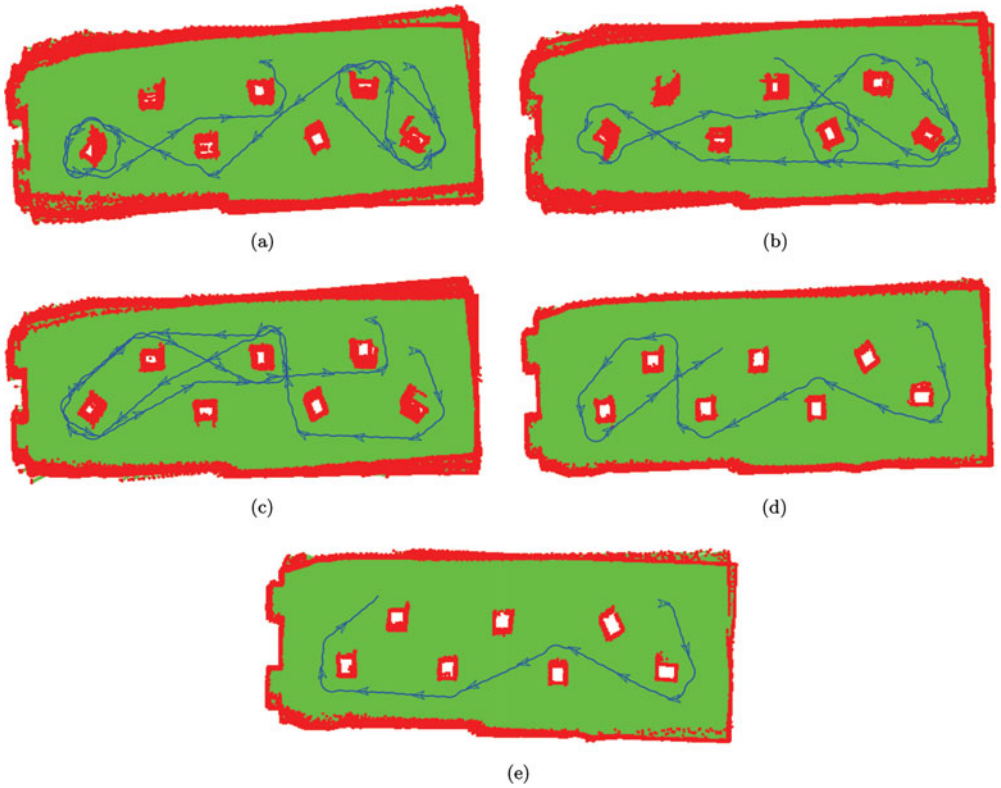


Fig. 31. The generated maps with different probability q_0 and different map building completing time t_f. (a) $q_0 = 0.3$ and $t_f = 313.6$ s. (b) $q_0 = 0.4$ and $t_f = 298.6$ s. (c) $q_0 = 0.5$ and $t_f = 336.2$ s. (d) $q_0 = 0.6$ and $t_f = 174.2$ s. (e) $q_0 = 0.7$ and $t_f = 130.2$ s.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

computationally efficient. Moreover, unlike many other publications in the area, a non-holonomic robot model was considered in the proposed method. Finally, the performance of the proposed algorithm is proved by mathematically rigorous analysis. The usefulness of the method in real applications is based on the fact that the proposed algorithm is computationally simple and easy to implement. Moreover, the proposed mathematically rigorous theoretical framework describes cluttered environments in which a good performance of the algorithm will be guaranteed.

While the simulations and experiments have confirmed the performance and advantages of the proposed algorithm, there are still some limitations on our method. In particular, in the proposed method, only the range finder sensor is considered. In the future work, the proposed methodology will be generalized and extended to different types of sensors. Moreover, the algorithm will be extended to 3D environments. An interesting direction of our future research will be to find a method to select a better value of the parameter q_0 in the step **A3** of the algorithm depending on the environment to be explored. Implementation of the proposed method in real applications will be another important direction of our future research. Potential real applications include environment map building in planetary development, seabed map building in oceans, and surveillance and patrolling in sensitive and contaminated regions.

Acknowledgment
This work was supported by the Australian Research Council.

References

1. M. Al Khawaldah and A. Nüchter, "Enhanced frontier-based exploration for indoor environment with multiple robots," *Adv. Robot.* **29**(10), 657–669 (2015).
2. A. AlDahak, L. Seneviratne and J. Dias, "Frontier-Based Exploration for Unknown Environments Using Incremental Triangulation," *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linköping, Sweden, IEEE (2013) pp. 1–6.
3. S. Almansa-Valverde, J. C. Castillo and A. Fernández-Caballero, "Mobile robot map building from time-of-flight camera," *Expert Syst. Appl.* **39**(10), 8835–8843 (2012).
4. A. Babinec, F. Duchoň, M. Dekan, P. Pásztó and M. Kelemen, "VFH* TDT (VFH* with Time Dependent Tree): A new laser rangefinder based obstacle avoidance method designed for environment with non-static obstacles," *Robot. Auton. Syst.* **62**(8), 1098–1115 (2014).
5. N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for searching environments in rescue operations," *Auton. Robots* **31**(4), 401–417 (2011).
6. J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. Robot. Auton.* **12**(6), 869–880 (1996).
7. J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.* **4**(1), 25–30 (1965).
8. R. G. Colares and L. Chaimowicz, "The Next Frontier: Combining Information Gain and Distance Cost for Decentralized Multi-Robot Exploration," *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, Pisa, Italy, ACM (Apr. 2016), pp. 268–274.
9. N. Doh, H. Choset and W. K. Chung, "Accurate Relative Localization using Odometry," *Proceedings of the IEEE International Conference on Robotics and Automation,* , Taipei, Taiwan, IEEE (2003) pp. 1606–1612.
10. N. L. Doh, H. Choset and W. K. Chung, "Relative localization using path odometry information," *Auton. Robots* **21**(2), 143–154 (2006).
11. L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Am. J. Math.* **79**(3), 497–516 (1957).
12. L. Freda and G. Oriolo, "Frontier-Based Probabilistic Strategies for Sensor-Based Exploration," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, IEEE (2005) pp. 3881–3887.
13. J. de Hoog, S. Cameron and A. Visser, "Autonomous Multi-Robot Exploration in Communication-Limited Environments," *Proceedings of the 11th Conference Towards Autonomous Robotic Systems (Taros 2010)*, Plymouth, UK, University of Plymouth, School of Computing and Mathematics (2010), pp. 68–75.
14. M. Hoy, A. S. Matveev and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica* **33**(3), 463–497 (2015).
15. C. Hsu, H. E. Chang and Y. Lu, "Map Building of Unknown Environment using PSO-Tuned Enhanced Iterative Closest Point Algorithm," *Proceedings of the International Conference on System Science and Engineering (ICSSE)*, Wroclaw, Poland, IEEE (2013), pp. 279–284.
16. S. Jain, S. Nandy, G. Chakraborty, C. S. Kumar, R. Ray and S. N. Shome, "Error Modeling of Laser Range Finder for Robotic Application using Time Domain Technique," *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Xi'an, China, IEEE (2011), pp. 1–5.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press

17. E. K. Kim, H. Cho, E. Jang, M. K. Park and S. Kim, "Map Building of Indoor Environment using Laser Range Finder and Geometrics," *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Busan, Korea, IEEE (2015) pp. 1259–1264.
18. Y. Kim and S. Kwon, "A heuristic obstacle avoidance algorithm using vanishing point and obstacle angle," *Intell. Serv. Robot.* **8**(3), 175–183 (2015).
19. W. Klingenberg," *A Course in Differential Geometry*, vol. 51, Springer Science & Business Media, Springer-Verlag New York, New York, USA (2013).
20. T. Liu and D. M. Lyons, "Leveraging area bounds information for autonomous decentralized multi-robot exploration," *Robot. Auton. Syst.* **74**(Part A), 66–78 (2015).
21. Y. Liu and Y. Sun, "Mobile Robot Instant Indoor Map Building and Localization using 2D Laser Scanning Data," *Proceedings of the International Conference on System Science and Engineering (ICSSE)*, Coventry, UK, IEEE (2012) pp. 339–344.
22. I. R. Manchester and A. V. Savkin, "Circular-navigation-guidance law for precision missile/target engagements," *J. Guid. Control Dyn.* **29**(2), 314–320 (2006).
23. A. S. Matveev, A. V. Savkin, M. Hoy and C. Wang," *Safe Robot Navigation among Moving and Steady Obstacles*, Elsevier, Butterworth-Heinemann, Oxford, UK (2015).
24. A. S. Matveev, H. Teimoori and A. V. Savkin, "A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance," *Automatica* **47**(3), 515–524 (2011).
25. A. S. Matveev, C. Wang and A. V. Savkin, "Real-time navigation of mobile robots in problems of border patrolling and avoiding collisions with moving and deforming obstacles," *Robot. Auton. Syst.* **60**(6), 769–788 (2012).
26. C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppé, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe, D. Duggins and J. Gowdy, "Moving object detection with laser scanners," *J. Field Robot.* **30**(1), 17–43 (2013).
27. S. W. Noh, N. Y. Ko and J. H. Han, "Integrating Elementary Functions for Autonomous Navigation of a Mobile Robot," *Proceedings of the 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Kuala Lumpur, Malaysia, IEEE (2014) pp. 591–593.
28. R. O'Flaherty and M. Egerstedt, "Optimal Exploration in Unknown Environments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, IEEE (2015) pp. 5796–5801.
29. S. Oßwald, M. Bennewitz, W. Burgard and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robot. Autom. Lett.* **1**(2), 716–723 (2016).
30. G. Pudics, M. Zsolt Szabó-Resch and Z. Vámossy, "Safe Robot Navigation using an Omnidirectional Camera," *Proceedings of the 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, Hangzhou, China, IEEE (2015) pp. 227–231.
31. R. Ray, V. Kumar, D. Banerji and S. N. Shome, "Simultaneous Localisation and Image Intensity Based Occupancy Grid Map Building–A New Approach," *Proceedings of the 3rd International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Kingston, Canada, IEEE (2012) pp. 143–148.
32. K. Rebai, A. Benabderrahmane, O. Azouaoui and N. Ouadah, "Moving Obstacles Detection and Tracking with Laser Range Finder," *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Munich, Germany, IEEE (2009) pp. 1–6.
33. C. T. Sauer, H. Brugger, E. P. Hofer and B. Tibken, "Odometry Error correction by Sensor Fusion for Autonomous Mobile Robot Navigation," *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference (IMTC 2001)*, Budapest, Hungary, IEEE (2001) pp. 1654–1658.
34. A. V. Savkin and M. Hoy, "Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments," *Robotica* **31**(2), 323–330 (2013).
35. A. V. Savkin and H. Teimoori, "Bearings-only guidance of a unicycle-like vehicle following a moving target with a smaller minimum turning radius," *IEEE Trans. Autom. Control* **55**(10), 2390–2395 (2010).
36. H. Teimoori and A. V. Savkin, "A biologically inspired method for robot navigation in a cluttered environment," *Robotica* **28**(5), 637–648 (2010).
37. J. Vallvé and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," *Robot. Auton. Syst.* **69**, 68–79 (2015).
38. C. Wang, A. S. Matveev, A. V. Savkin, R. Clout and H. T. Nguyen, "A semi-autonomous motorized mobile hospital bed for safe transportation of head injury patients in dynamic hospital environments without bed switching," *Robotica* **34**(8), 1880–1897 (2016).
39. D. Wang, Y. Duan and J. Wang, "Environment exploration and map building of mobile robot in unknown environment," *Int. J. Simul. Process Modelling* **10**(3), 241–252 (2015).
40. M. Wang, W. Wang, J. Xiong and L. Yan, "A Consistent Map Building Method Based on Surf Loop Closure Detection," *Proceedings of the IEEE 3rd Annual International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)*, Nanjing, China, IEEE (2013) pp. 92–95.
41. T. Wattanavekin, T. Ogata, T. Hara and J. Ota, "Mobile robot exploration by using environmental boundary information," *ISRN Robot.* **2013**, 1–11 (2013).
42. J. Xue, L. Zhang and T. Grift, "Variable field-of-view machine vision based row guidance of an agricultural robot," *Comput. Electron. Agriculture* **84**, 85–91 (2012).
43. M. Yokoyama and T. Poggio, "A Contour-Based Moving Object Detection and Tracking," *Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, IEEE (2005) pp. 271–276.

https://doi.org/10.1017/S0263574717000194 Published online by Cambridge University Press