International Journal of Intelligent Robotics and Applications (2021) 5:558–575
https://doi.org/10.1007/s41315-021-00199-8

REGULAR PAPER

A novel solution with rapid Voronoi-based coverage path planning in irregular environment for robotic mowing systems

Kuo-Chun Huang[1] · Feng-Li Lian[1] · Chien-Tung Chen[2] · Chung-Hou Wu[2] · Chao-Cheng Chen[2]

Received: 30 October 2020 / Accepted: 10 August 2021 / Published online: 31 August 2021

Abstract
This paper presents a novel solution of coverage path planning for robotic mowing applications. The planning algorithm is based on the Boustrophedon motions and the rapid Voronoi diagram. The coordinate conversion and the sweeping vector is defined by minimum bounding box and the Voronoi travel paths are designed to reduce the computational cost and execution time compared to conventional heuristic methods. The tracked path is controlled via dynamic feedback in standard lawn mowing robots under robot operating system (ROS). Within ROS, the information exchanging among different tasks of both extensive simulation cases and experimental field tests can be conducted easily. When meeting unknown obstacles, the proposed algorithm can re-plan its paths dynamically. The performance of the proposed algorithm is compared to several conventional coverage algorithms in terms of time efficiency, coverage, repetition, and robustness with respect to both concave and convex shapes. The field tests are conducted to demonstrate that the applicability of the sensor fusion and robustness of the proposed algorithm for the complete coverage tasks by robotic mowers.

Keywords Lawn mowing robot · Autonomous system · Coverage path planning · Voronoi diagram · Obstacle avoidance

1 Introduction

Due to the increasing dependence on robotics, autonomous lawn mowers are becoming popular in public or private green space. In 2019, the market growth of the global robotic lawn mowers was observed at US $28.5 billion (Grand View Research 2020), and estimated to keep growing at more than 15% between 2017 and 2023 (Report Buyer 2020).

With regard to the senor used on mower, many of early commercial mower using a shallow-buried wire for confining random motion robotic mowers (Patterson et al. 2019). This kind of coverage approaches rely on randomized components and heuristics computation which are not efficient (Sportelli et al. 2020). To overcome these issues, thus, in this paper, sensor fusion algorithm based on extended Kalman filter (EKF) for the estimation of ground vehicle dynamics is used. For global localization, the real-time kinematic global positioning system (RTK-GPS) should be used (Li et al. 2018). On the other hand, for the local localization, the odometry and inertial measurement unit (IMU) are used together. The ultrasonic sonar sensor might be used for detecting unknown obstacles around the robot.

Without random walking approach, the core work of lawn mower systems aims at developing an efficient coverage path planning. Coverage path planning (CPP) is a classical problem in robotics and a task of determining a path that passes through all the points on the area of interest in a systematic way while avoiding any collision with other obstacles. Existing complete coverage algorithms using cellular decomposition of the free space (Choset and Pignon 1998), which resolve the free space into simple cells. After the decomposition, each cell would then be covered by zig-zag motion (Acar et al. 2003). Efficient algorithms for the lawn mowing problem by using zig-zag motion are discussed in Arkin et al. (2000). Nevertheless, not many studies have been reported on the approach to reduce the computation time as well as to decrease the execution time of path generation.

In this paper, the total solution for automatic lawn mower is introduced, such as the transformation among different coordinate systems, the conversion of global positioning system (GPS) and Universal Transverse Mercator (UTM), which is

✉ Feng-Li Lian
fengli@ntu.edu.tw

[1] Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan

[2] URS Robot Inc., Taipei 114, Taiwan

adopted from equal earth map projection suggested by Bojan et al. (2019). For finding the optimal solution of sweeping direction, the minimum bounding box (MBB) is used to reduce turning waypoints (Freeman and Shapira 1975). For complete coverage path planning, the paths with high coverage rate from a high-resolution grid-based map is used, and the turning window is updated at each step. As the unknown obstacles detected by the mower, the modified Bug algorithm (Kamon et al. 1998) is operated to re-plan the coverage paths. When the robot encounters blind alley situation, the proposed algorithm based on the Boustrophedon motions and the rapid Voronoi diagram (namely, B-RV) will check for the region to determine whether it has not been cleaned or not. After that, the robot will travel to the next area using the Voronoi graph, which dwindles the computation complexity, and reduces the execution time to coverage by picking up less number of waypoints. Finally, extensive software simulations on several random obstacles and experimental field tests for mower navigation are presented along with the error-feedback controller on motors.

In summary, main contributions presented in this paper are described as follows:

- The whole solution of the automatic mowing system is illustrated, including innovative map projection, optimal sweep direction observing, Boustrophedon motion for coverage path planning, and online re-planning planner based on modified Bug algorithm.
- The adaptation of the Voronoi-based path planning algorithm as a topological planning is proposed. In comparison to traditional methods, the path provided by the proposed methodology (B-RV) is optimal not only in computation time but also in execution time due to the decision of path waypoints to obtain higher efficiency.
- The navigation in non-convex environment for the robotic lawn mower based on low cost sensor fusion is presented in commercial product under ROS is provided.

The paper is organized as follows. A brief review of related works regarding mower platforms and the planning skill is discussed in Sect. 2. Section 3 describes the problem formulation of coverage task. Section 4 discusses the proposed idea, method, and approach. Section 5 presents simulation cases with evaluation. Section 6 demonstrates the experimental field tests of grass sculpture by one standard lawn mowing robot. Finally, Sect. 7 concludes the paper and outlines future work.

2 Literature survey

2.1 Sensors used on mowing systems

Autonomously performing coverage tasks on large outdoor areas is required for robotic mowing system. Thus, several different modes for optimal path planning, including minimal working time, minimal energy consumption, and mixed operation are proposed (Shiu and Lin 2008). The advantages of such system aims at reducing labor cost, saving time and energy from users, preventing operators from possible injuries, and allowing the owner to maintain the grassland at any time on any day. In order to replace the human labor, the sensors used on mowing systems should be chosen. The sensors such as, RTK-GPS, odometry, IMU, etc., are discussed in the following.

2.1.1 Real-time kinematic global positioning system (RTK-GPS)

Various autonomous systems have been developed with shallow-buried boundary wires, such as Bosch "Indego" and Toadi Orders' "Toadi", utilizing random walk for tackling the mission. Initially, the region of interest is defined by a shallow-buried boundary wire, which helps the mower to detect electro-magnetic field. By hitting the boundary wire, the machine can stop and change heading direction until it hits the wire again. Although mowing along this trajectory can finish the task finally, this kind of operation will generate continual overlapping when the lawn size is huge and it might take a longer time to complete the task (Sportelli et al. 2020). Besides, the buried boundary wire can damage the lawn and take a lot of effort to setup. To prevent the problem, it is good to provide a guidance system in advance. For current lawn mower market, Ambrogio L60 announces their robots use GPS data instead of installing a perimeter wire. Nonetheless, GPS data is not suitable for precise localization near trees and other obstructions due to the dependence on several observed satellites, as studied in Nourani-Vatani et al. (2006). Moreover, a guidance system is not provided in the product. Thus, in this paper, RTK-GPS is used to localize the mower in the global space and offers an accuracy range, which can replace the function of using buried boundary wire and enhance the precision of position data.

2.1.2 Odometry

Traditional lawn mowers are driven along the buried boundary wire while measuring steps and movements with the wheel odometry (Rottmann et al. 2020). As usual, for low-cost localization, the wheel odometry is used for define the mower in the local map.

2.1.3 Ultrasonic sensor

One of the intuitive methods to identify the obstacle is vision-based detection by equipped camera(s) (Fukukawa et al. 2016). For this approach, it utilizes a texture classification with the collection of a large number of images. Then, the distribution of grass model can easily obtain by any type of learning algorithms. The accuracy of leaning high dimensional data is required, which is onerous to test in practical applications (Guo et al. 2017). Not only the high computation loading on the mower occurs, but it is also expensive to implement on commercial products. Therefore, the ultrasonic sensor is adopted to measure the distance to any objects. The sensing data can be used to avoid collisions with stones or obstacles lying on the lawn and to prevent possible damage to the robot. The ultrasonic sensor can offer a reliable detection method under control system.

2.1.4 Inertial measurement unit (IMU)

The most frequently used sensors on mobile robot are IMU ones. With inertial sensors the mower position and velocity can be computed by integrating the acceleration measurements. However, the errors of the position and the orientation grow over time due to the accumulation of the measurement noise and bias (Cong and Fang 2007). Thence, any type of sensor fusion is required.

2.1.5 Sensor fusion

Based on global positioning system, the sensor fusion is proposed in Wasif (2011). Combining with sonar sensor to identified obstacles, a conceptual framework based on semi-autonomous lawn mowing systems is outlined in Patterson et al. (2019), which help the developers to construct a system with engineering perspective on the design of user-integrated, semi-autonomous mowing systems. Nevertheless, the demonstration and the experiment analysis are not discussed in recent study. Moreover, the solution of the whole process is not provided in recent papers as well.

In this paper, the odometry, RTK-GPS, IMU data are fused together by EKF. Odometry and the IMU errors can be reduced by comparing the results with an absolute position by RTK-GPS. Consequently, the complete solution for the robot lawn mower is implemented in under ROS by using the robot localization package (Moore and Stouch 2016). Finally, the mower utilizes an error-feedback controller to generate motor commands to execute the planned trajectory based on proposed B-RV motion planner.

2.2 Path planning

In order to handle the mower motion, the global path planning before the mowing task plays an important role in the whole system. In the past two decades, a lot of researches have gone into the development of CPP, which is the task of determining a best path that can cover all of the specific region (Galceran & Carreras 2013). However, most of studies only demonstrate the coverage tasks for cleaning robots in small indoor region, seldom research focus on large outdoor coverage scenarios. Thus, this paper aims to develop a rapid coverage algorithm, and implement in outdoor robotic lawn mower.

In the majority of the related studies, the practical constraints such as power supply should be considered, initial position of robots usually assumed to be near the boundary (Shnaps and Rimon 2016). Several rules for designing the CPP algorithms are summarized in Cao et al. (1988), such as simple trajectories like straight lines should be used. Therefore, the proposed B-RV algorithm is developed based on Boustrophedon motion. There are a few categories of CPP methods described as follows. The exact cellular decomposition methods can be divided into two parts by the trapezoidal partition, IN and Out, which can be executed in offline (Oksanen and Visala 2009). Each cell can connect through the edges of the graph by using the stack, then the coverage problem is reduced to determine a walk through the graph. Sometimes, when the situation is complicated, the trapezoidal decomposition may generate too many cells, which can be merged together to form much bigger cells intuitively. Therefore, to overcome this problem, the Boustrophedon Cellular Decomposition (BCD) can be used (Choset and Pignon 1998). In BCD, Critical points of a Morse function (Milnor et al. 1969) are used to determine the cell decomposition. However, if the robot stays in rectilinear environments, all the critical points that produced by the Morse functions should be degenerate. Thus, the contact-based coverage of rectilinear environments is proposed in Butler et al. (2002). Among several CPP methods, the most common way is the grid-based method. An offline procedure by wave front propagation is presented in Zelinsky et al. (1993). The algorithm first assigns 0 to the goal and then assigns 1 to all its surrounding cells. Next, the repetition of algorithm stops when the start cell is reached.

Nonetheless, when the map is quite large or there are numerous obstacles on the environment, the CPP step would take high computational cost and memory consumption (Chen and Zhu 2019). In order to solve the problem of complete coverage path planning, Viet et al. (2012) proposed an efficient coverage approach for cleaning robots based on Boustrophedon motions and A* algorithm (BA*). The approach is efficient with the coverage path length and total number of Boustrophedon cells formation. However, at each backtracking point the robot takes a long time to choose the shortest collision-free path as the travel path which cause inefficient in terms of time complexity.

Recently, there is another path planning method, two-way proximity search (TWPS), based on Boustrophedon motion, which is combined with advanced two-way proximity search from point to point in order to reduce the distance of the backtracking path (Khan et al. 2017). This method brings about a lower overlap rate and less computation time when comparing with other complete coverage path planning (CCPP) algorithms based on Boustrophedon motion. Nevertheless, the algorithm still uses heuristic approach which needs to find the backtracking path multiple times in a large environment. It wastes a lot computational power and takes much time to define trajectories.

In this paper, the proposed motion planner not only improves the inefficient exploration in travel path, but also capable of re-planning the coverage path for detecting unknown obstacles online. Besides, the efficiency of travel time during the robot navigation is improved. Finally, the solution for robotic lawn mower is discussed and analyzed in detail. The proposed B-RV algorithm is applied to deal with the real-world mowing issues in autonomous mowing system.

3 Problem formulation

The focus of the paper is on the coverage path planning of mower robots with kinematic constraints and simultaneously avoiding obstacles. The coverage radius of the robot is denoted as r, which is assumed to be the resolution of grid map. The region of interest (ROI) consists of the accessible region, Ra, and the obstacle region, Ro, and $Ro \cap Ra = \emptyset$ (Tang et al. 2020). The workplace is denoted as $\mathcal{W}$. An example of target region is given in Fig. 1. The slash area represents the obstacle region.

The mower robot and obstacles are three-dimensional objects in its workspace. Nonetheless, the motion of robot is constrained in a two-dimensional plane $\mathcal{R}^2$. Therefore, the mower robot and obstacles can be projected onto the x–y plane and considered as two-dimensional objects in the workspace $\mathcal{W} = \mathcal{R}^2$ (Choi et al. 2017).

In order to specify the position of the robot on the plane, the relationship between the global reference frame of the plane and the local reference frame of the robot is characterized as the definition of related parameters shown in Fig. 2.

Since the vehicle operates under differential drive., as a result, the vehicle position, (x, y), in the world coordinate corresponds to the center of the vehicle. The heading angle, θ, is defined as the angle between the heading orientation of the vehicle and the x-axis of the world coordinate. The symbol, k, is used to defined the time step. The configuration $q(k)$ of the robot is defined as follows:

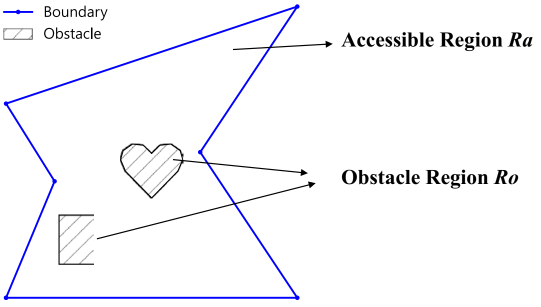


Fig. 1 An example of target region

$$q(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} \tag{1}$$

For the rotational motion shown in Fig. 2b, the angle, δ, is defined as the angle between the heading orientation of vehicle and the orientation of the target pose (Siegwart et al. 2011). Then, the next configuration $q(k+1)$ can be characterized as follows:

$$q(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) + \delta \end{bmatrix}. \tag{2}$$

If δ is positive, the mower robot will rotate counterclockwise, and vice versa.

For the straight-ahead motion shown in Fig. 2c, if the mower robot moves forward about distance d, the next configuration $q(k+1)$ can be characterized as follows:

$$q(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) + d\cos(\theta(k)) \\ y(k) + d\sin(\theta(k)) \\ \theta(k) \end{bmatrix}. \tag{3}$$

Furthermore, for the easy of presentation, the following conditions are used:

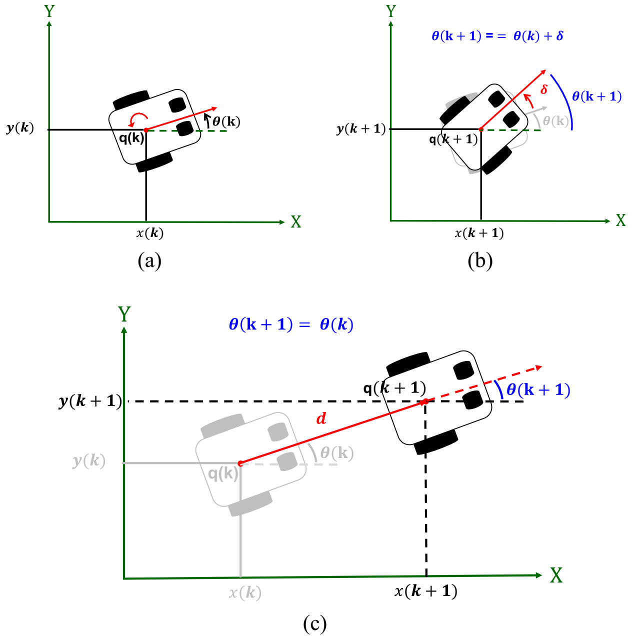


Fig. 2 The configuration of the mower in workspace *W*. **a** The configuration of the mower at step k. **b** The configuration of the mower at step k + 1 after rotating. **c** The configuration of the mower at step k + 1 after going straight ahead

1. The GPS signal is fixed.
2. The position information of the region boundary is available in advance.
3. Assume that the fields are flat and there exist trivial obstacles in accessible region.

4 Proposed idea, method and approach

The proposed B-RV algorithm is implemented on the same high-resolution grid map where the cell size is set to the size of the sweeping tool. The Voronoi map representation is used where the cell size is independent of sweeping coverage map. The implementation of Voronoi map is discussed in Section IV-D.

The localization information by RTK-GPS are adopted as precise sensing data. In order to accelerate the whole process, the rapid processing of conversion between the GPS data and the UTM data become important in the system. The coordinate transformation, i.e., the Equal Earth map projection (Bojan et al. 2019), is adopted which is a new equal-area pseudo cylindrical projection for world maps.

The overall process is illustrated in Fig. 3. At first, the GPS signal is converted to the UTM coordinate. Then, the MBB is applied to find the longest vector and to ensure that the full coverage matching the required longest vector. After approximating the grid-map from the boundary and obstacles, the grid map is initially filled with zero as unexplored cell. The robot mower records each grid and label the current position. If no unexplored cell is found, the overall coverage task ends. However, the connection of each region with collision-free region should be taken into consideration. Thus, the Voronoi travel map needs to be set up. After then, the mower robot will travel to the next start point by the Voronoi path and the Dijkstra algorithm. The detailed descriptions of the projection of GPS data, the sweeping vector selection, coverage path generation scheme based on grid, Voronoi travel path algorithm, and re-planning mechanism will be described in the following subsections.

4.1 Converting GPS signal to UTM coordinate

Since the GPS-based navigation system is used on the robotic lawnmower, the data should be continuously converted into an equal map or converted back during each movement. Therefore, the detail of conversion from the raw data (latitude and longitude) will be discussed in this section.

The UTM coordinate system is the geographical latitude and longitude system which is expressed in the two-dimensional projection of the surface of earth. The surface of the Earth is cut into 60 spindles of 6 degrees each.In the proposed process, the Equal Earth map projection (Bojan et al. 2019) is applied, which is a new equal-area pseudo cylindrical projection for the world map. Such like the Robinson projection, the projection equations are not only fast to evaluate but also easy to implement.

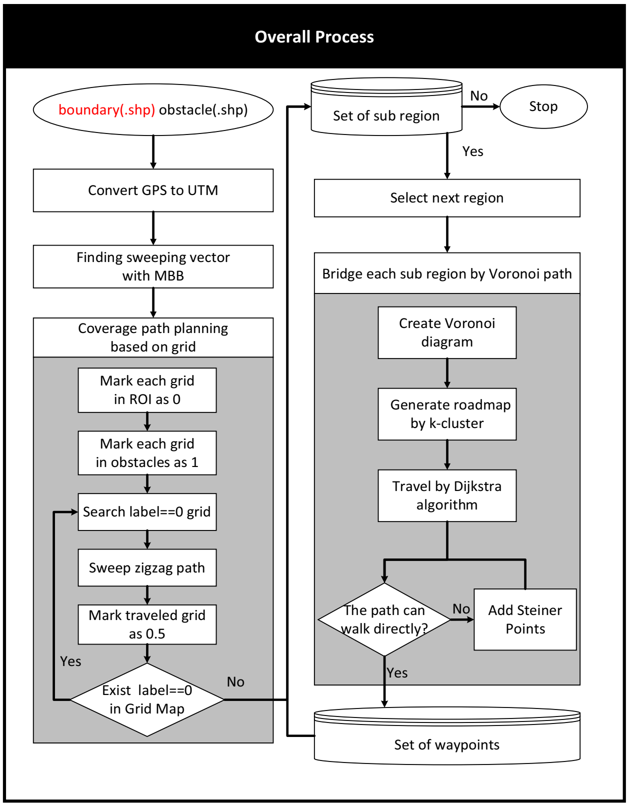

Overall Process
boundary(.shp) obstacle(.shp)
Convert GPS to UTM
Finding sweeping vector with MBB
Coverage path planning based on grid
Mark each grid in ROI as 0
Mark each grid in obstacles as 1
Search label==0 grid
Sweep zigzag path
Mark traveled grid as 0.5
Yes
Exist label==0 in Grid Map
No
Set of sub region
No
Stop
Yes
Select next region
Bridge each sub region by Voronoi path
Create Voronoi diagram
Generate roadmap by k-cluster
Travel by Dijkstra algorithm
The path can walk directly?
No
Add Steiner Points
Yes
Set of waypoints

Fig. 3 Overall process of the proposed coverage path planning algorithm

The features of the equal earth projection include:

1. Straight parallel lines on the map are easily understood and able to compare how far the places from the equator.
2. All of the meridians are uniformly spaced along any line of latitude.
3. It is easy for the developer to implement the conversion formulas in software as shown in Eqs. (4), (5), (6), which are formulated in Bojan et al. (2019). Then, the program can be executed efficiently. Equation (4) represents the unit sphere with respect to latitude ϕ. Equations (5) and (6) derive the conversion of coordinate frame from GPS to UTM.

$$sin\theta = \frac{\sqrt{3}}{2} sin\phi, \tag{4}$$

$$x = \frac{2\sqrt{3}\cdot\lambda\cdot cos\theta}{3\cdot\left(A_1 + 3\cdot A_2\cdot\theta^2 + \theta^6\left(7\cdot A_3 + 9\cdot A_4\cdot\theta^2\right)\right)}, \tag{5}$$

$$y{=}\theta \cdot \left(A_1{+}A_2 + \theta^2 + \theta^6 \cdot \left(A_3 + A_4 \cdot \theta^2\right)\right), \tag{6}$$

where λ, ψ refer to the longitude and latitude coordinates, respectively, θ is a parametric latitude, and A_1 to A_4 are related polynomial coefficients defined as: $A_1 = 1.340264$, $A_2 = -0.081106$, $A_3 = 0.000893$, and $A_4 = 0.003796$.

4.2 Finding the sweeping vector with minimum bounding box (MBB)

In this approach, MBB is used to denote the smallest enclosing box of the set of points and to represent the region of interest boundaries. Since the result of its MBB is the same as the MBB of its convex hull, it may be used heuristically to speed up the computation time by making convex hull first (Toussaint 1983).

The technique for computing MBB consists of two steps. In the first step, the convex hull is computed. If the polygon is convex already, this procedure can be ignored. There are numerous algorithms for computing convex hulls, such as Quick Hull (Barber et al. 1996), gift wrapping (Jarvis 1973), Graham's algorithm (Graham 1972). In the second step, the rotating calipers method is applied to compute the resulting MBB. At first, the rotating calipers are invented to calculate the diameter of convex polygons. Besides, it can be used to compute the minimum and maximum of the convex polygon.

Through rotating each edge of convex hull polygon, the candidates of the four lines of the rectangle area can be found. This process is repeated until the period occurred. The resulting MBB is the candidate with the smallest area. Figure 4 shows the convex hull and the MBB of the vertices of a concave polygon. Algorithm 1 describes the pseudocode for the MBB method.

Algorithm 1: Sweeping Direction Determining by Minimum Bounding Box (MBB)

```
Input: points
Output: vector
convex_poly ← convex_hull(points)
min_area = ∞
min_rotation_matrix ← []
for each edge ∈ convex_poly do
    angle ← atan2(edge)
    rotation_matrix ← $\begin{bmatrix} cos\ (angle) & -sin(smgle) \\ sin\ (angle) & cos\ (angle) \end{bmatrix}$
    rot_points ← rotation_matrix x convex_poly
    area ←Area(rot_points)
    if area < min area then
        area < min ← area
        vector ← coincided edge vector of convex poly
    end if
end for
return vector
```

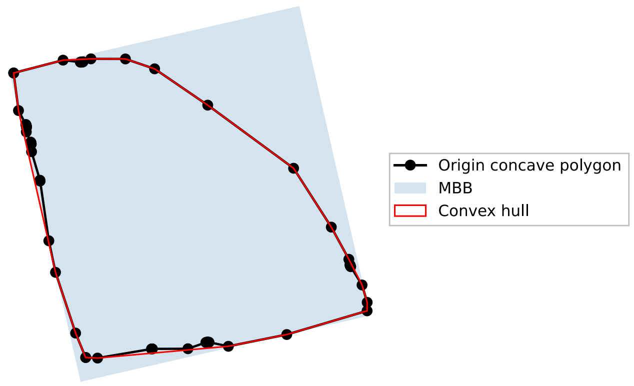


Fig. 4 Convex hull and the MBB of concave polygon

4.3 Grid-based coverage path planning

Before the path planning, the idea to discretize the map into many grids from Dakulović et al. (2011) is applied. The whole environment is divided into several square cells with given specific resolution size, represented a list of G elements, P = {1, ..., G}. All the information can be interpreted as the 2D array and used to keep each of the center point of the grid $c_i \in \mathfrak{R}^2, i \in \mathrm{P}$ in memory. In the proposed method, a sampling-based planner works by searching the graph. Since the graph consists of the spatial nodes, the mower robot can observe its location at any time.

Each node can be denoted as one of three possible states s_{node}. Hence, the node on the map is coded by the following equation:

$$s_{node}{=}\begin{cases} 1, & \text{unvisitable} \\ 0.5, & \text{visited} \\ 0, & \text{unvisited} \end{cases}. \tag{7}$$

The initial status of the grid map can be visualized as Fig. 5a. The grid connectivity is constructed by the von Neumann neighborhood where the mower robot moves to the adjacent nodes in four directions, namely, left, right, up and down, in each step. The mower robot will start at the corner of the longest edge, then the mower robot has the ability to perform the traveling on each grid horizontally until hitting the black cell. The grid that the mower robot traveled will be marked as 0.5 and plot as gray cell, as shown in Fig. 5b.

Since the goal of the mower robot is to travel all the grid in map. The turning window is defined in Eq. (8) to make mower robot move to next grid instead of hitting the black cell.

$$\text{turning\_window} = [\text{moving\_direction}, \text{sweeping\_direction}) \tag{8}$$

The parameters of moving_direction (− 1,0,1) determines the heading on the map of each line. Meanwhile, the parameter of sweep_direction (− 1,0,1) determines the parallel vector of each line on the map. Therefore, the coverage path uses a back and forth motion as perpendicular rows to the sweep direction as shown in Fig. 6.

4.4 Path planning based on Voronoi diagram

4.4.1 Create the roadmap from Voronoi diagram

Initially, the concept from approximating the obstacles with points on the boundary edges is used to generate the Voronoi diagram of the approximating points (Sugihara 1993). The Voronoi diagram is a spatial partition structure generally used in many science domains. In the Voronoi diagram, the space can be divided into n regions corresponding to given n points, which are called the Voronoi cells. That is, each center point of the Voronoi cell is the nearest neighbor of its corresponding Voronoi cell's surrounding points. After creating the Voronoi diagram within the map, there might exist many edges crossing the obstacle and boundary which are not expectable as shown in Fig. 7a. Since the traveling via the Voronoi points and by Dijkstra algorithm will encounter the problem. Thus, the removal of the edges that cross obstacles from this diagram is necessary (Bhattacharya and Gavrilova 2008). Figure 7b shows the resultant Voronoi road map.

4.4.2 Determining the backtracking point

After creating a series of Voronoi points, determining the shortest path from one node to another in a graph is an issue. In most cases, the coverage algorithm cannot complete with a single Boustrophedon path planning. Each Boustrophedon motion starts at a position S and ends at a position E. The new starting point s_{new} of Boustrophedon path can be at any uncovered position in the accessible region. Nevertheless, randomly choosing the next starting point may fragment the nearest path, then the total path length will be higher. Therefore, the best starting point should be chosen by covered region. Given a list of R regions, $R = \{1, ..., R\}$, the Euclidean distance is adopted as a greedy strategy to establish the travel path between $\{[E_1, S_2], [E_2, S_3], [E_i, S_{i+1}]\}$, where $i \in R$. The cost of the path length is defined as follows:

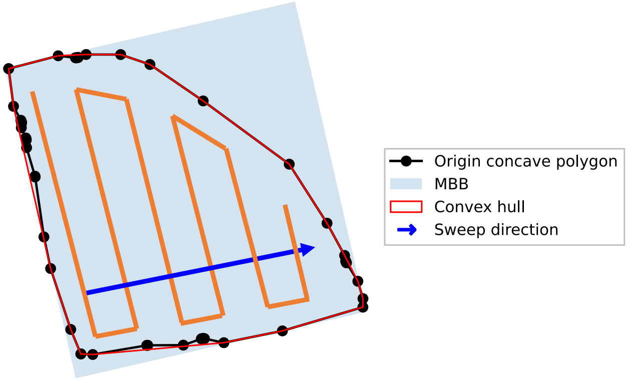


Fig. 6 Illustration of the sweep direction

$$s_{new} = \underset{s\in R}{argmin}(f(s, s_c)), \tag{9}$$

where $f(s, s_c)$ is a cost function that defines the distance between an element s in the backtracking list R and the critical point s_c is the critical point of each region.

In general, if there are no obstacles as a barrier interrupt the path, the Euclidean distance is used to calculate the distance as shown in Fig. 8a. On the other hand, under the carrier condition shown in Fig. 8b, the relationship of l_{se} is described as follows:

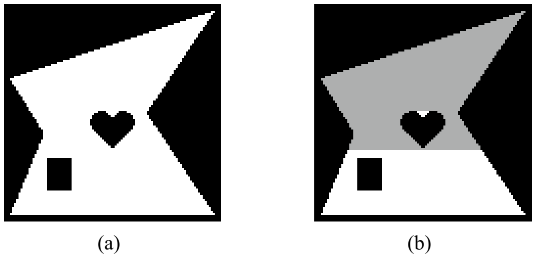


Fig. 5 The grip map approximation. **a** The grid out the boundary or inside the obstacle filled as 1(black). **b** The visited grid can be represented as 0.5 (gray)

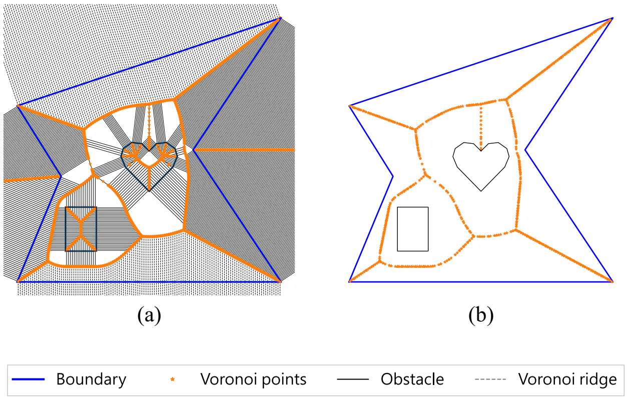


Fig. 7 Roadmap generation using Voronoi diagram. **a** The edges cross obstacles and out of boundary. **b** Removal the Voronoi points which is not in accessible region

$$l_{se}=\begin{cases}\|p(s)-p(e)\|, & \text{if collision - free}\\ l_{se}^{Voronoi}, & \text{if collision - on path}\end{cases} \tag{10}$$

4.4.3 Bridge each region with Voronoi roadmap

The rapid Voronoi roadmap is solved with the Dijkstra algorithm. In order to speed up calculation time, the idea in Niu et al. (2019) is applied to construct the roadmap with the kd-Trees from Voronoi points before Dijkstra search. The kd-tree (Bentley 1975) is a special case of binary space partitioning tree, which organizes points in a k-dimensional space with a spatial data structure. The detail description of roadmap construction is shown in Algorithm 2.

Algorithm 2: Roadmap Generation.

```
Input:  $Q$ : Voronoi Points
        $O$: Obstacle Points
        $N\ KNN$: Number of closet neightbors
Output: $G = (V, E)$ : Voronoi roadmap
$V \leftarrow null$, $E \leftarrow null$
while $|V| < N\ KNN$ do
$q \leftarrow$ a random collision-free in $Q$
$V \cup \{q\}$
end while
for each $q$ in $V$ do
$N \leftarrow$ the $N\ KNN$ closest neighbors from $q$ in $V$
for each $q_N$ in $N$ do
  if $(q,q_N)$ !∈ $E$ and the local planner $(q,q_N)$ != $null$
then
      $E \leftarrow E \cup \{q,q_N)\}$
    end if
  end for
end for
return
```

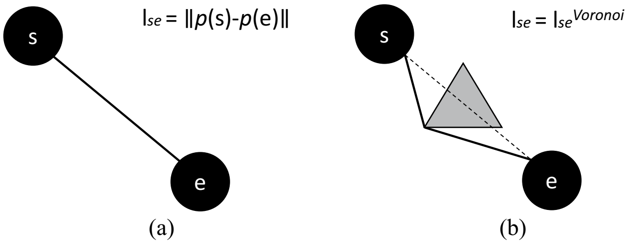


Fig. 8 **a** lse obtained by Euclidean distance under collision-free conditions. **b** lse obtained by Voronoi diagram under collision conditions

Algorithm 3: Dijkstra Algorithm.

```
Input: $G = (V, E)$ :
Output: $s$[]: status, $p$[]: predecessor, $d$[]: distance
for each $v$ in $V$ do
    $d[v] \leftarrow \infty$
    $p[v] \leftarrow null$
end for
$d[s] \leftarrow 0$
$S \leftarrow null$
$Q \leftarrow V$
while $Q$ != $null$ do
  u←Extract_Min(Q)
  S←S∪{u}
end while
for each $edge(u,v)$ do
  if $d[u] + w[u,v] < d[v]$ then
     $d[u] \leftarrow d[u] + w[u,v]$
     $p[v] \leftarrow u$
  end if
end for
```

The N_KNN value in the roadmap generation algorithm is empirically chosen as 10. The edge will be connected by the Voronoi points which are useful for searching in the next step. The Dijkstra's Algorithm is a weighted breadth-first search, which calculates the shortest path between two nodes of the graph G. Denote each node as $v \in V$, where V is the set of nodes. Every node has status $s(v)$: non-visited, visited, or labeled, and the predecessor node is recorded as $p(v)$. Furthermore, $d(v)$ is used to represent the distance from the start node. The initial process, all nodes are marked as non-visited, *p(v)* is all equal to null, and $d(v)$ is assigned as ∞. The whole process of the Dijkstra's Algorithm is shown in Algorithm 3.

4.5 Replanning mechanism based on modified Bug algorithm

In order to avoid the robot limited to static environments, the re-planning mechanism is considered in this section. To overcome the unknown obstacle issue, the modified Bug algorithm is used in this research. Bug algorithms show a good framework to reconcile reactivity and convergence which are simple obstacle avoidance algorithms that ensure that any accessible target can be reached (Kamon et al. 1998). In this paper, the modified Bug algorithm is based on Bug-1 algorithm and called as "modified Bug algorithm". During start-to-goal in each Boustrophedon motion, the mower moves along the path toward goal until it either encounters the goal or an obstacle. If the robot encounters an unknown obstacle, let q_H be the hitting point where the robot detects an obstacle. Then, the robot waits for 30 s and circumnavigates the obstacle until it returns to q_H. The main difference with respect to other traditional Bug algorithm (Kamon et al. 1998) is that it is possible to stop and recreate the coverage path instead of repeatedly finding the leaving point q_L and continues onward towards the goal as shown in Fig. 9. The detail description of modified Bug algorithm is shown in Algorithm 4.

Algorithm 4: Modified Bug algorithm

```
1: while not $q == q_{goal}$ do move towards the goal
2: if hit an obsatcle then
3:   mark $q$ as $q_H$
4:   circumnavigate it
5:   if $q == q_H$ then
6:     replan the coverage paths
7:       break
8:   end if
9: end if
10: end while
```

The exit condition for boundary following in modified Bug algorithm is describe as below.

Given the location of robot q in workspace $\mathcal{W} = \mathcal{R}^2$, and the rays radially emanating from sensor with θ, then the sensing distance function between obstacle and the saturated distance function are defined as follows:

$$l(q,\theta) = dist\left(q, q + \lambda \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}\right). \tag{11}$$

$$l(q,\theta) = \begin{cases} l(q,\theta) if\, l(q,\theta) < R \\ \inf, otherwise \end{cases} \tag{12}$$

where λ represents the sensing range of the robot, and R represents maximum sensing radius.

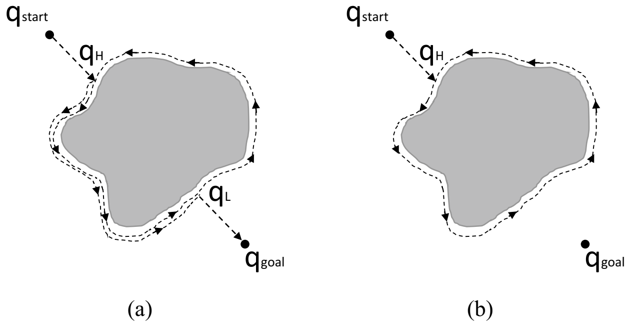


Fig. 9 Blocking example of Bug algorithm applied to irregular environments and our modification proposal. **a** The Bug-1 algorithm **b** the modified Bug-1 algorithm

5 Simulation studies

To assess the performance of the proposed method in terms of computational efficiency and effectiveness, related simulations are performed on a Core i7-7700 M CPU 3.6 GHz with 16 GB RAM computer. The simulations are conducted in the Gazebo simulator. Several computational geometry libraries from Shapely are used. The mower robot is created based on a differential-drive vehicle model to simulate simplified vehicle dynamics. These simulation tasks are described in the following subsections, and all the map is assumed smaller than a square with side length 10 m. Also, the mower robot is modeled by a circle with a radius of $r = 0.5$ m. The mower robot is put on the flat ground without any pothole or small steep slope.

All the simulations are tested in Gazebo using ROS messages (Noori et al. 2017). The sonar rays can be seen in the simulation world which is able to identify its surroundings in unknown maps as shown in Fig. 10. The whole process is calculated in an interpreted language, Python.

5.1 Demonstration of the proposed algorithm

The proposed method uses the zigzag coverage path indicated by the solid lines with different color in order to represent the time domain. For example, the mower will start at the red color and end up with the purple. In the other words, the red points are the region that mower first travel, the purple points are the region that mower last travel. The mower follows the colorful zigzag coverage path and mows.

5.1.1 Re-planning mechanism

The first simulation aims to verify the complete coverage of proposed algorithm. A boundary with the coordinate list: [(0,0), (0,10), (10,10), (10,0), (0,0)] is used. Figure 11a shows the reference trajectory of the mower robot in the first simulation. The mower robot starts at Start to perform the coverage task, then it encounters the unknown obstacle. The robot pauses at coordinate (5.1,3) for about 30 s in Fig. 11b, and performs the modified Bug algorithm around the obstacle with the rectangle coordinate list: [(3.0, 3.0), (5.0, 3.0), (5.0, 6.0), (3.0, 6.0), (3.0,3.0)] in Figs. 11c, 13e, where the blue solid line represents the modified Bug path. Figure 11d illustrates the Replanned motion. After then, the mower robot recreates the coverage path based on B-RV and keep working under collision-free conditions in Fig. 11e, as discussed in Section IV.E. In the end of the process, the mower robot will keep execute the Boustrophedon motion to the position End in Fig. 11f.

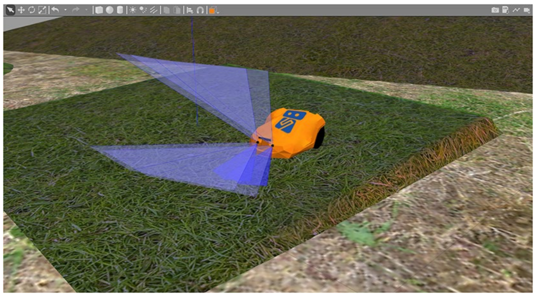


Fig. 10 The simulated scene and vehicle in the Gazebo simulator

In this simulation, the Boustrophedon motion can be simple and regular in a simple boundary and obstacle. The result of the coverage rate of the robot in Fig. 11f is 99.6%.

In order to compare the impact of the modified Bug algorithm on the difference of the reference paths in unknown obstacle map. Two snapshots of the re-planned path are shown in Fig. 12. Regardless of whether there exists unknown obstacle in the figure, the solid gray line represents the constant path. Figure 12a displays the robot will pass through the blue solid line if there is no predefined obstacle in the map. In contrary, if the robot encounters the unknown obstacle, the robot will avoid the obstacle based on blue solid line as indicated in Fig. 12b.

5.1.2 B-RV algorithm

For the simulation of coverage algorithms, three test environments of irregular shape were constructed as shown in Fig. 13. Three different kind of maps consisting of diverse shapes and several obstacles, including circles, triangle, rectangle, pentagon, heart-shape, etc. Each of the map is composed of concave angles and is capable of representing the real-world examples. The area of obstacles is from 10 to 20% of the workspace. The white space denotes free regions.

In order to test the performance of the proposed algorithm for a lawn mower with arbitrarily shaped obstacles and boundary. Besides, the efficiency of Voronoi road map should also be presented. The second workspace of a boundary with the coordinate list: [(0,0), (1, 2.5), (0,4.0), (6,6), (4,3), (6,0), (0,0)] is used. The mower robot is placed at the bottom-right corner of the workspace as well. The map is designed with both convex polygon and concave polygon. Apart from adding a simple rectangular obstacle, without taking the rectangle obstacle away, the heart shape obstacle that contains the curve is added to show the algorithm has ability to perform well both in convex and concave scenes.

Figure 14 shows the resulting paths of simulations. Figure 14a shows the paths generated by the Boustrophedon motion and the Voronoi roadmap and Fig. 14b shows the total path connected from starting point Start to ending point

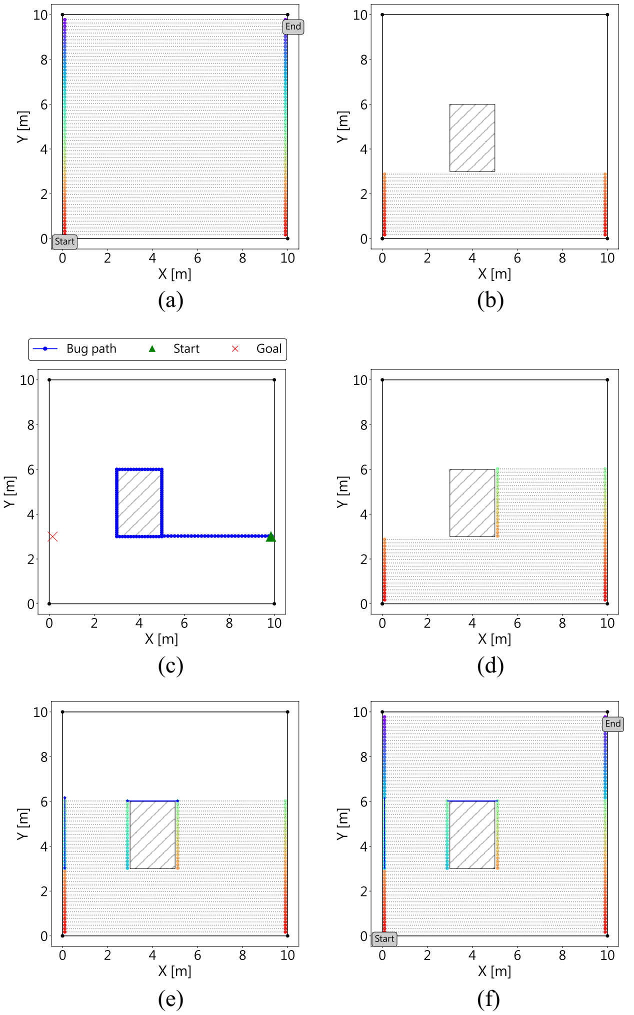


Fig. 11 Proposed of replanning CPP based on Bug algorithms. **a** Reference paths are marked as different colors. **b** The robot encounters the unknown obstacle. **c** Create the modified Bug path and label the location of obstacle. **d** Following the coverage paths after replanning. **e** Recreate the coverage paths based on B-RV. **f** Illustration of the result paths

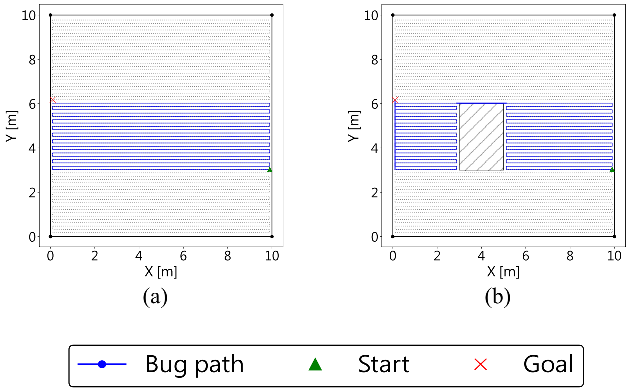
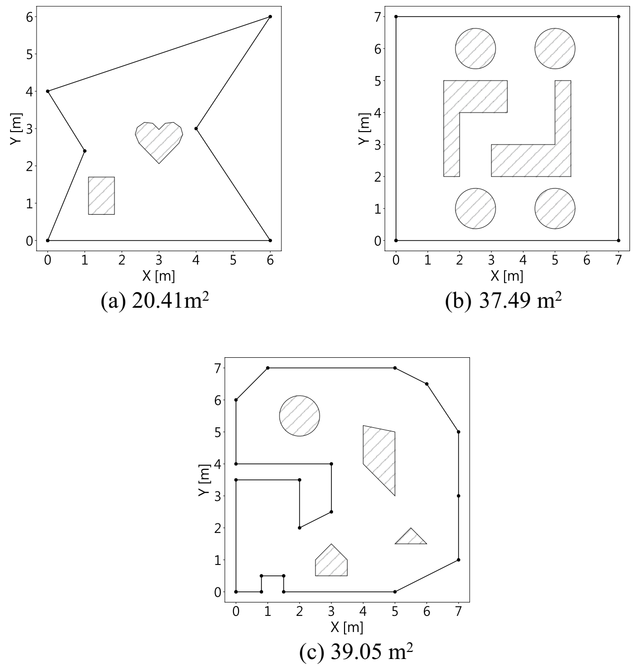


Fig. 12 The difference of the reference paths. **a** Before detecting unknown obstacle. **b** After detecting unknown obstacle

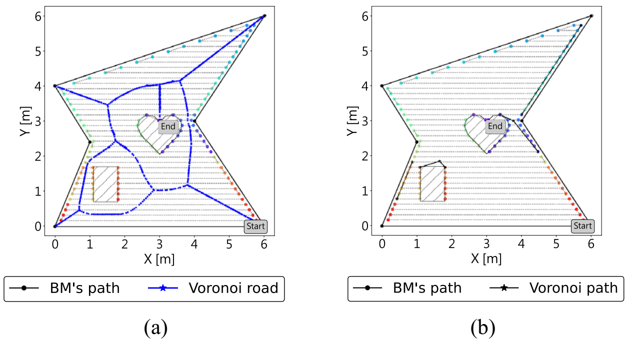


(a) 20.41m^2

(b) 37.49 m^2

(c) 39.05 m^2

Fig. 13 Test environment



(a)

(b)

Fig. 14 The coverage paths are obtained from proposed algorithm. **a** Boustrophedon paths are marked as different colors. **b** The travel paths are connected by Voronoi roadmap

End which covers the whole space. The concave corner is the interesting issue, which lets the mower robot generate another more Boustrophedon path and result in creating a new traveling path. The solid lines marked by asterisks depict the linking paths between two successive cells. The calculation time required to cover generate the workspace of Fig. 14 is within in 3 s. Moreover, the more efficiency metrics are compared in Section V.B. These simulation result show that the robot has ability to cover almost all of the free regions of the workspace, except for some small portions around the concave shape of the obstacle, such as the top of the heart, which is a fragment region in the workspace. The coverage rate is defined as the ratio of the area calculated by approximate grid cell and the real accessible region, then the successful coverage rates of the robot in Fig. 14b is 99.51%.

5.2 Performance comparison

In this section, the performance of the proposed B-RV algorithm from the perspective of computational efficiency and execution effectiveness are analyzed. For fairness, the area of region of interest of the mower robot is the same in each pair of comparison, and the comparison of each algorithm is accomplished by Boustrophedon motion. The result of the proposed B-RV algorithm is compared with that of the conventional CPP method: BCD (Choset & Pignon 1998), BA* (Viet et al. 2012) and TWPS (Khan et al. 2017). These methods determine the next uncleaned region, and apply different path planning techniques. For example, BCD which decomposes the map into non-overlapping cells. BA* uses the Boustrophedon motions and A* algorithm to obtain the shortest and collision-free trajectory. For instance, given a consistent heuristic $h(s)$, the A* search just expands the vertices whose total cost, $f(s) = g(s) + h(s)$, is less than c. It takes a long time for calculations. TWPS utilizes use a two-way proximity search. In this study, the less waypoints for robot to chase by using the proposed B-RV algorithm results in the shorter execution time in whole mowing process. Moreover, the proposed B-RV algorithm reduce the time for calculating a path instead of using a lot of heuristic techniques. This research implements the pseudocode of conventional CPP methods including BCD, BA*, and TWPS.

During the coverage process, apart from verifying the computation time required between the different algorithm, the efficiency is indeed a major issue during the coverage process. We evaluated the simulation in three kind of maps from Fig. 13 with five performance measures as below:

$$\begin{aligned}&\text{Normalized computation time to coverage}\\&\quad=\text{computation time}/\text{covered area}\end{aligned}\tag{13}$$

$$\text{Coverage} = (\text{covered area}/\text{free area}) \times 100\tag{14}$$

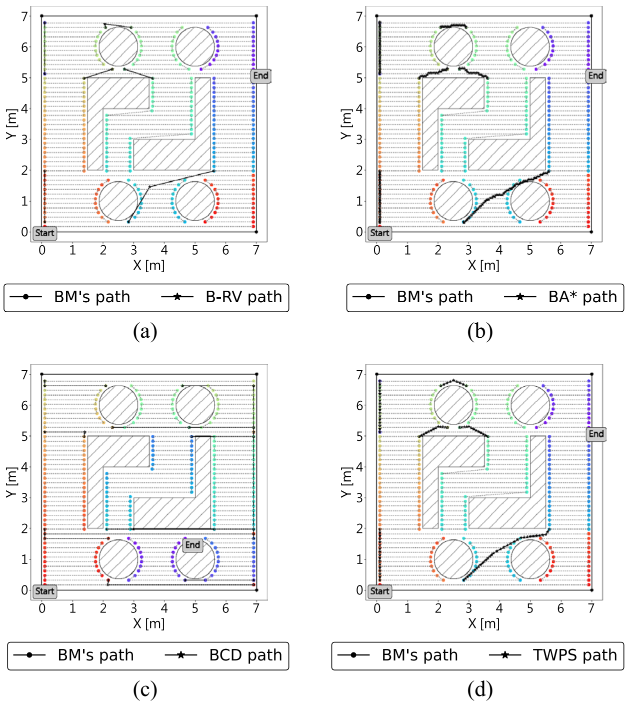
$$\begin{aligned}&\text{Normalized execution time to coverage}\\&\quad=\text{execution time}/\text{covered area}\end{aligned}\tag{15}$$

$$\text{Total travel distance} = \Sigma_{k=1}^{n-1}\left|(x,y)_k-(x,y)_{k+1}\right|\tag{16}$$

$$\text{Path repetition rate} = \text{total travel distance} * \text{resolution}/\text{covered area}\tag{17}$$

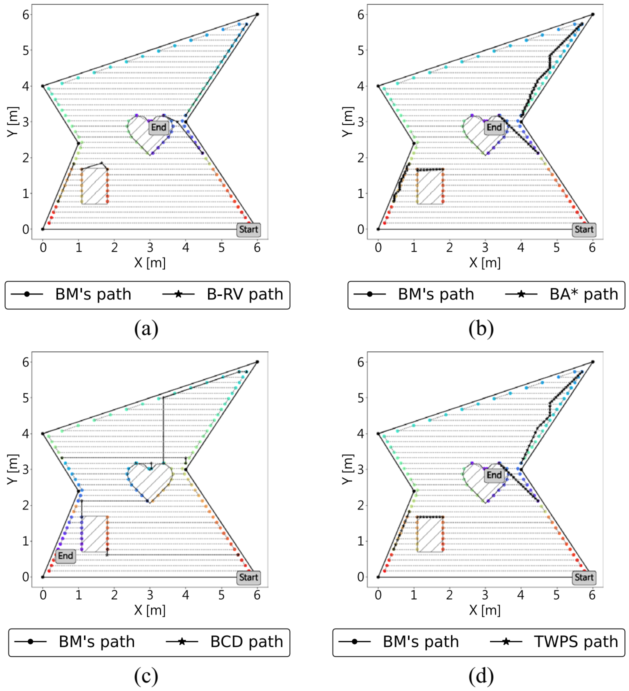
The results of the coverage paths are shown in Figs. 15, 16 and 17. Each of the map we take three-time simulations for each algorithm. Their statistics are shown in Table 1. We can easily find that the proposed algorithm is the best in all performance measures. As to normalized computation time to coverage, BA* resulted in the worst performance because it requires a high computational memory as the map become larger. BCD is better than TWPS, since it just follows the priority in the stack and utilizes the Depth First Search (DFS) based optimization of coverage sequence to move to the last backtracking point. Analysis of total coverage rate clearly indicates that the TWPS resulted in the worst performance because the expansion is constrained to a neighbor tile. Figure 15d shows the coverage missing on the top of the heart by TWPS algorithm.

The proposed B-RV algorithm also shows a lower normalized execution time to coverage and a less total travel distance because the generated path which exists less waypoints and straightly path than the others. In irregular environments, BA* and BCD can result in not only more sharp turns but a greater number of waypoints compared to the proposed algorithm, as can easily be seen from the trajectories in Fig. 16. The black asterisk points on the coverage maps indicate that it consumes much time at the turning points. The acceleration constraints limit the translational and rotational accelerations between two neighboring points. By using the proposed B-RV algorithm, the fewer points on the travel path are used and, thus, the B-RV takes less time to finish the process than the other algorithms do. For path repetition rate, BCD shows the highest rate. By reason of the Voronoi map we used, the mower has ability to travel in safe middle road, avoiding crash into the obstacle or falling out the boundary. Some missed area when mowing can be compensated when traveling.



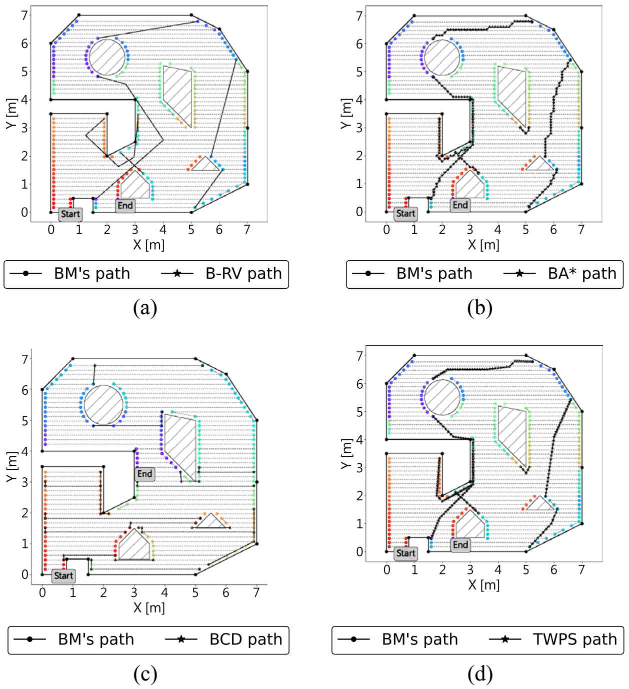
(a) (b) (c) (d)

Fig. 16 Simulation of different CPP methods in map (**b**). **a** B-RV. **b** BA*. **c** BCD. **d** TWPS



(a) (b) (c) (d)

Fig. 15 Simulation of different CPP methods in map (**a**). **a** B-RV. **b** BA*. **c** BCD. **d** TWPS



(a) (b) (c) (d)

Fig. 17 Simulation of different CPP methods in map (**c**). **a** B-RV. **b** BA*. **c** BCD. **d** TWPS

Table 1 Mean and std. dev. (N = 9) of five performance indices of the four algorithms under comparison in irregular shape environment

Quantity		Algorithm			
		BA*	BCD	TWPS	Proposed (B-RV)
Normalized computation time to coverage (s/m^2)	Mean	0.61	0.35	0.41	0.14
	Std.	0.02	0.0.2	0.01	0.01
Coverage (%)	Mean	98.3	98.4	95.2	98.4
	Std.	0.7	0.7	1.2	0.7
Normalized time to coverage (s/m^2)	Mean	28.6	32.6	27.5	27.3
	Std.	5.3	3.5	4.6	2.4
Total travel distance (m)	Mean	9.7	16.2	9.6	10.1
	Std.	0.1	0.1	0.1	0.1
Path repetition rate (%)	Mean	3.6	4.5	3.6	3.7
	Std.	0.1	0.2	0.1	0.1

As to standard deviations of the various quantities, the proposed algorithm accomplished the lowest values thus demonstrating robustness of the algorithm against irregular environment shape or an arbitrary scheme of map.

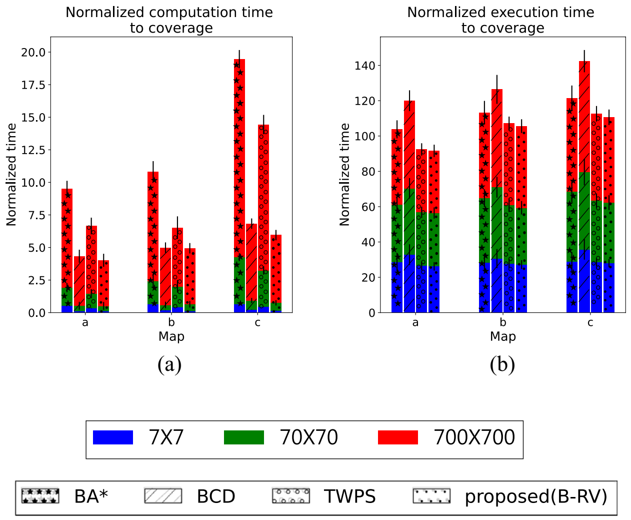
In order to verify the ability of proposed algorithm, the computation time and execution time for each method on the 7 × 7, 70 × 70 and 700 × 700 map sizes are represented by overlaid blue, green and red bars, respectively in Fig. 18.

The bar which was hatched with the dotted is the proposed B-RV algorithm, and the other which hatch with empty is BA*. No matter in Normalized computation time to coverage or in Normalized execution time to coverage, the time consuming achieved by the proposed B-RV algorithm is always smaller than that achieved by BA*. Under the statistical view, the proposed B-RV algorithm doubtless improves the efficiencies of the coverage path planning. The computation time and execution time of 9 simulations increases as the maps become larger.

Overall, the simulation results show that the computation time of the proposed algorithm is shorter than those of other methods. The proposed B-RV algorithm effectively deals with the challenges of irregular environments and overcome the complete coverage problem of autonomous mower robots in workspaces because the proposed method can plan the coverage path with a reduced execution time and planning the path rapidly than conventional method.

From the simulation results, it is observed that the coverage path achieved by the proposed B-RV algorithm is much better than by other approaches because of the following three reasons:

1. The proposed B-RV algorithm guides the mower robot to the next uncovered region based on the Voronoi map which just need to calculate only first time as the robot initial. On the contrary, BA* using the shortest collision-free path found by A* search for each time of traveling, the heuristic algorithms are essentially methods by trial and error. TWPS also used the heuristic approach. BCD always generates the most total number of cells to travel. Thus, for the other approaches will take a lot of time to find the optimal shortest path, i.e. when the grid map resolution doubles, the computation and memory requirements increase four times.
2. The execution time to coverage required for the proposed B-RV algorithm is smaller than the others. Since the B-RV pick up the less waypoints, which reduce the accumulated time of visiting the certain cell. For BA* and TWPS algorithm, they just take the shortest path into account to travel to the next region when encounter the blind alley, thus the mower robot should visit more relay point along the boundary, which brings about increasing the accelerations instead of reducing the average velocity for faster coverage. For BCD, it takes the longest time to cover the whole region due to the higher travel distance.



(a) (b)

Fig. 18 **a** The normalized computation time to coverage time of 9 simulations in each scenario. **b** The normalized execution time to coverage time of 9 simulations in each scenario

3. The result of coverage rate and the path repetition rate achieved by the proposed B-RV algorithm is the almost same or less than the other three algorithms use Boustrophedon motions as the way of the ox to accomplish the basic technique in order to cover workspaces.

6 Experiment results

With the aim of evaluating practicality and applicability of the proposed coverage path planning strategy, the overall framework is tested in the grass field on the off-the-shelf mower robot shown in in Fig. 19. The mower robot is round shape with cutter radius 0.07 m and fitted with an array of sensors and two differential-driven motors. The sonar sensors are used to detect the unknown factors. The mower may stop in front of unidentified obstacles for 30 s and determine the new path in real-time. All the coded programs are implemented and executed under the standard robotics middleware, ROS (Noori et al. 2017), where most of the low-level device controls and common functionalities can be easily implemented with the message-passing procedure between processes.

On top of the ROS, the proposed coverage algorithm is coded with Python, and run on a laptop (Intel i5-10210U, 1.60 GHz). The sensor fusion process is followed by standard EKF-based localization algorithm (Moore and Stouch 2016). In the prediction step, the state of the robot is estimated by the odometry and the IMU. The predicted state is then corrected by a RTK-GPS measurement.

Furthermore, key features of the mower robot are summarized as follows:

1. Two wheel encoders, which can measure wheel rotation with the optical encoders.
2. A RTK-GPS receiver, which provides geolocation and time information anywhere on Earth.
3. A 9-DOF WT931 IMU (accelerometer + gyroscope + magnetometer), which gives the robot orientation as a yaw angle to control the heading of robot.
4. Two HC-SR04 ultrasonic distance sensors, which measure distance and detect the presence of an object without making physical contact.

Considering the covering range of the cutter radius and also for in coverage completion, a square region of 0.15 m by 0.15 m is covered by a cutter at a given time. Thus, the resolution is set as 0.15 m which is equal to the cutter radius of the mower robot. The values of the basic parameters are set as follows: the robot translational velocity and angular velocities are set to 0.6 m/s and 1.0 rad/s.

For controlling the robot in whole process, each target point in waypoints stack can be obtained by that discussed in Section IV.

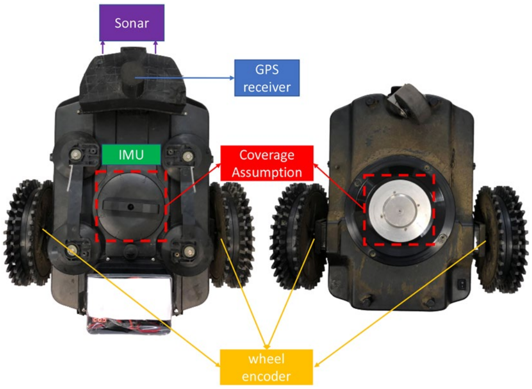


Fig. 19 Configuration of our differential drive based lawn mower robot equipped odometry, GPS, IMU and sonar sensors

The sensors on the robot are capable of identifying its location. Therefore, the next configuration $q' = \begin{bmatrix} x' & y' & \theta' \end{bmatrix}$ can be determined by Eqs. (2) and (3). The angle and the distance of d can also calculate by current position and target position under the ROS controller.

For experimenting coverage algorithms, three practical environments of irregular shape are constructed as shown in Fig. 21, which shows that this algorithm can be used for more practical applications as shown in Fig. 13. That is, instead of the ideal shape using in simulation, a more complicated icon, such as “TAIPEI CITY” words, robot, and cake are tested and exists more concave shape.

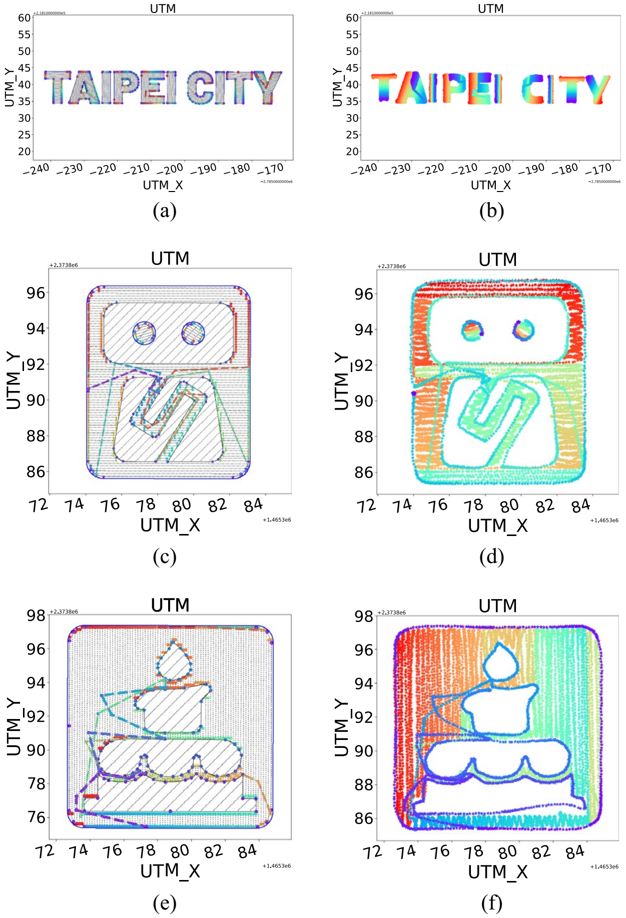
The generated paths of “TAIPEI CITY”, robot, cake simulated in Gazebo are shown in Fig. 20a, c, e, respectively. Figure 20b, d, f show the result of the three experiments with recorded RTK-GPS data. In Fig. 20a, the combination of Boustrophedon motions and the Voronoi-Dijkstra search path with multiple concave corners is used. The result of “TAIPEI CITY” icon shows successful of linked path. In Fig. 20c, as a means of displaying the adaptability and capability of the proposed algorithm in narrow space, a cut robot logo is tested. The mower robot performs nine Boustrophedon motions to cover the entire workspace, seven paths for covering the body and two paths for accomplishing the eyes. Since the above experiments barely generated a high precision scenario, to test the more complicated geometry, the cake icon is tested as shown in Fig. 20e. The result in Fig. 21c shows that it is same as expect in simulation.

The snapshots of experimental results are taken by the camera on the flying drone in Fig. 21. Eventually, we analyze snapshot results. It is obvious that the proposed algorithm is capable of accomplishing the experiment field with specific icons, and respect to the top of view, all the cases are very similar to the simulation results, which validates our approach for a real system in an irregular environment. In addition, the proposed algorithm provides a high coverage rate and effectively demonstrate the mowing task.

In summary, adaptability of our approach to irregular environments are demonstrated. The whole solution can be applied in the real-ground. The experiments show that although the mower is equipped with only RTK-GPS, IMU, odometry, and sonar sensor without camera and bury-wired, B-RV algorithm finishes the grass carving and fulfills the coverage tasks under the rapid computation time and execution time. The solution minimized calculation time at each iteration, so that the total operation time is ultimately minimized. The waypoints are minimized due to using the Voronoi-based algorithm so that the execution time also be minimized.

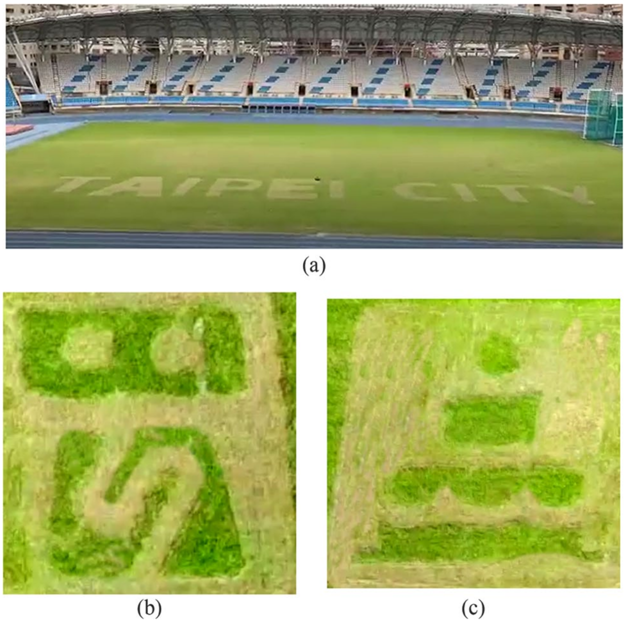
7 Conclusions and future work

In this paper, a novel solution of coverage algorithm is presented for robotic mower system. The proposed algorithm is effortless to implement under standard sensor fusion and able to solve any scenario. The ROS middleware is also applied to communicate the message in system. The robots are designed according to the mower-based approach, that is, the localization signal, such as RTK-GPS is a fundamental requirement for a robot so that the robot is capable of knowing where it is. Simultaneously, the design of coverage regions via the Boustrophedon motions in an incremental manner. Furthermore, to travel to the closest unvisited region, the robot is equipped with a brilliant tracking mechanism. If a robot reaches the end point of a Boustrophedon motion, it performs three main tasks, namely, (1) detecting each point based on the grid map of the workspace; (2) determining the shortest traveling path through the Dijkstra search algorithm combining with the Voronoi diagram; (3) following the Voronoi roadmap path to cover the next unvisited region, which decreases the computation time and execution time.



(a) (b) (c) (d) (e) (f)

Fig. 20 The reference paths for robot lawn mower and the results of experiments are shown. **a** The reference path of "TAIPEI CITY" icon. **b** The RTK-GPS signal of "TAIPEI CITY" icon's coverage task. **c** The reference path of robot icon. **d** The RTK-GPS signal of robot icon's coverage task. **e** The reference path of cake icon. **f** The RTK-GPS signal of cake icon's coverage task



(a) (b) (c)

Fig. 21 The snapshots taken by drone's camera. **a** The top view of "TAIPEI CITY" icon. **b** The top view of robot's icon. **c** The top view of cake's icon

From the experimental result, it has little limitations on arbitrary geometry and excellent robustness with respect to the 3D geometry shape of environments. Owing to real-world geometry is not flat due to several small potholes in the grass, some parts of the icon are not as clear as simulation shape. Thus, the path traversed by the mower may be crooked, such as the right eye of the robot shown in Fig. 20b and the bottom of the cake shown in Fig. 20c. It is possible to deal with 3D field terrain geometry issues in the future works.

In summary, the proposed (B-RV) provides an efficient way for the users of automated lawn mowers to develop their system and implement their jobs in real world. Ongoing work will focus on applying the B-RV to the scenarios of 3D terrains and keep optimizing the path length of the Voronoi traveling path.

Besides, the path repetition rate of the coverage path planning will be minimized for deeper analyses to future work.

Acknowledgements This research is supported by the Ministry of Education, Taiwan,under Grants: NTU-110L890603, by the Ministry of Science and Technology, Taiwan, under Grants: MOST 109-2221-E-002-168, MOST 110-2221-E-002-168, MOST 110-2634-F-002-038,and by Industrial Technology Research Institute, Hsinchu, Taiwan.

References

Acar, E.U., Choset, H., Zhang, Y., Schervish, M.: Path planning for robotic demining: robust sensor-based coverage of unstructured environments and probabilistic methods. Int. J. Robot. Res. **22**(7–8), 441–466 (2003)

Arkin, E.M., Fekete, S.P., Mitchell, J.S.B.: Approximation algorithms for lawn mowing and milling. Comput. Geom. **17**(1–2), 25–50 (2000)

Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. ACM Trans. Math. Softw. **22**(4), 469–483 (1996)

Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. Assoc. Comput. Mach. **18**(9), 509–517 (1975)

Bhattacharya, P., Gavrilova, M.: Roadmap-based path planning—using the Voronoi diagram for a clearance-based shortest path. IEEE Robot. Autom. Mag. **15**(2), 58–66 (2008)

Bojan, Š, Tom, P., Bernhard, J.: The Equal Earth map projection. Int. J. Geogr. Inf. Sci. **33**(3), 454–465 (2019)

Butler, Z.J., Rizzi, A.A., Hollis, R.L.: Contact sensor-based coverage of rectilinear environments. In: Proceedings of the IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics, Cambridge, MA, USA, pp. 266–271 (2002)

Cao, Z.L., Huang, Y., Hall, E.L.: Region filling operations with random obstacle avoidance for mobile robots. J. Robot. Syst. **5**(2), 87–102 (1988)

Chen, M., Zhu, D.: Real-time path planning for a robot to track a fast moving target based on improved Glasius bio-inspired neural networks. Int. J. Intell. Robot. Appl. **3**(2), 186–195 (2019)

Choi, S.Y., Lee, S., Viet, H.H., Chung, T.: B-Theta*: an efficient online coverage algorithm for autonomous cleaning robots. J. Intell. Robot. Syst. **87**(2), 265–290 (2017)

Choset, H., Pignon, P.: Coverage path planning: the boustrophedon cellular decomposition. Field Serv. Robot. **31**, 203–209 (1998)

Cong, M., Fang, B.: Multisensor fusion and navigation for robot mower. In: Proceedings of 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO), Sanya, China, pp. 417–422 (2007)

Dakulović, M., Horvatić, S., Petrović, I.: Complete coverage D* algorithm for path planning of a floor-cleaning mobile robot. IFAC Proc. Vol. **44**(1), 5950–5955 (2011)

Freeman, H., Shapira, R.: Determining the minimum-area encasing rectangle for an arbitrary closed curve. Commun. ACM **18**(7), 409–413 (1975)

Fukukawa, T., Sekiyama, K., Hasegawa, Y., Fukuda, T.: Vision-based mowing boundary detection algorithm for an autonomous lawn mower. J. Adv. Comput. Intell. Intell. Inform. **20**(1), 49–56 (2016)

Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**(12), 1258–1276 (2013)

Graham, R.: An efficient algorithm for determining the convex hull of a finite planar set. Inf. Process. Lett. **1**(4), 132–133 (1972)

Grand View Research: Lawn mowers market size, share & trends analysis report by product (petrol, electric, manual, robotic), by end use (residential, commercial & govt.), by region (MEA, Asia Pacific, North America), and segment forecasts, 2020–2027, technical report, 2020. Industry Report Number GVR-1-68038-927-2. [Online] (2020). https://www.grandviewresearch.com/industry-analysis/lawn-mowers-market. Accessed 9 Aug 2020

Guo, H., Wu, X., Sun, J., Ou, Y., Feng, W.: Robust grass boundary detection for lawn mower with a novel design of detection device. In: Proceeding of IEEE International Conference on Real-Time Computing and Robotics (RCAR), Okinawa, Japan, pp. 218–222, (2017)

Jarvis, R.A.: On the identification of the convex hull of a finite set of points in the plane. Inf. Process. Lett. **2**(1), 18–21 (1973)

Kamon, I., Rimon, E., Rivlin, E.: TangentBug: a range-sensor-based navigation algorithm. Int. J. Robot. Res. **17**(9), 934–953 (1998)

Khan, A., Noreen, I., Ryu, H., Doh, N.L., Habib, Z.: Online complete coverage path planning using two-way proximity search. Intell. Serv. Robot. **10**(3), 229–240 (2017)

Li, T., Zhang, H., Gao, Z., Chen, Q., Niu, X.: High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMSIMU integration. Remote Sens. **10**(2), 205–226 (2018)

Milnor, J., Spivak, M., Wells, R.: Morse Theory, vol. 51. Princeton University Press, Princeton (1969)

Moore, T., Stouch, D.: A generalized extended Kalman filter implementation for the robot operating system. In: Proceedings of the 13th International Conference on Padua, Italy, pp. 335–348 (2014)

Niu, H., Savvaris, A., Tsourdos, A., Ji, Z.: Voronoi-visibility roadmap-based path planning algorithm for unmanned surface vehicles. J. Navig. **72**(4), 850–874 (2019)

Noori, F.M., Portugal, D., Rocha, R.P., Couceiro, M.S.: ROS based service robot platform. In: Proceedings of IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China (2017)

Nourani-Vatani, N., Bosse, M., Roberts, J.: Practical path planning and obstacle avoidance and autonomous mowing. In: Proceedings of the Australasian Conference on Robotics and Automation, Auckland, New Zealand, pp. 6–8 (2006)

Oksanen, T., Visala, A.: Coverage path planning algorithms for agricultural field machines. J. Field Robot. **26**(8), 651–668 (2009)

Patterson, A.E., Yuan, Y., Norris, W.R.: Development of user-integrated semi-autonomous lawn mowing systems: a systems engineering perspective and proposed architecture. AgriEngineering **1**(1), 453–474 (2019)

Report Buyer: Robotic lawn mower market–global outlook and forecast 2018–2023. https://www.reportbuyer.com/product/5398687/robotic-lawn-mower-market-global-outlook-and-forecast-2018-2023.html. Accessed 9 Aug 2020

Rottmann, N., Bruder, R., Xue, H., Schweikard, A., Rueckert, E.: Parameter optimization for loop closure detection in closed environments. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, pp. 36–43 (2020)

Shiu, B.-M., Lin, C.-L.: Design of an autonomous lawn mower with optimal route planning. In: Proceedings of IEEE International Conference on Industrial Technology, Chengdu, China, pp. 1–6 (2008)

Shnaps, I., Rimon, E.: Online coverage of planar environments by a battery powered autonomous mobile robot. IEEE Trans. Autom. Sci. Eng. **13**(2), 425–436 (2016)

Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: Introduction to Autonomous Mobile Robots, 2nd edn., pp. 58–60. MIT Press, Cambridge (2011)

Sportelli, M., Pirchio, M., Fontanelli, M., Frasconi, C., Martelloni, L., Caturegli, L., Gaetani, M., Grossi, N., Magni, S., Raffaelli, M., Peruzzi, A.: Autonomous mowers working in narrow spaces: a possible future application in agriculture? Agronomy **10**(4), 553–568 (2020)

Sugihara, K.: Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. IEEE Robot. Autom. Mag. **55**(6), 522–531 (1993)

Tang, Y., Zhou, R., Sun, G., Di, B., Xiong, R.: A novel cooperative path planning for multirobot persistent coverage in complex environments. IEEE Sens. J. **20**(8), 4485–4495 (2020)

Toussaint, G.: Solving geometric problems with the rotating calipers*. In: Proceeding of IEEE Melecon, vol. 83. Athens, Greece (1983)

Viet, H.H., Dang, V.-H., Laskar, M.N.U., Chung, T.: BA*: an online complete coverage algorithm for cleaning robots. Appl. Intell. **19**(2), 217–235 (2012)

Wasif, M.: Design and implementation of autonomous lawn-mower robot controller. In: Proceedings of International Conference on Emerging Technologies, ICET, Islamabad, Pakistan, pp. 1–5 (2011)

Zelinsky, A., Jarvis, R.A., Byrne, J.C., Yuta, S.: Planning paths of complete coverage of an unstructured environment by a mobile robot. In: Proceedings of International Conference on Advanced Robotics, pp. 533–538 (1993)

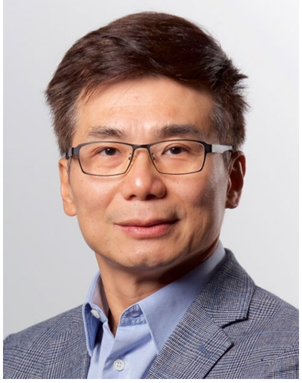
Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kuo-Chun Huang received the B.S. degree in Mechanical Engineering from National Chiao Tung University, Hsinchu, Taiwan in 2019, and the M.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 2021. His research interests include artificial intelligence, image processing, path planning and navigation for mobile robots.

Feng-Li Lian received the B.S. and M.S. degrees from National Taiwan University in 1992 and 1994, respectively, and the Ph.D. degree from the University of Michigan in 2001.Since 2002 he has been in the Department of Electrical Engineering, NTU, and since 2015 he also holds a joint appointment with Industrial Technology Research Institute, Hsinchu, Taiwan. From 2001 to 2002, he was a postdoctoral scholar at California Institute of Technology, and from 2009 to 2013, he was also the Division Director of Information Management, Computer & Information Networking Center, NTU. He is the recipient of the Youth Automatic Control Engineering Award from Chinese Automatic Control Society, Taiwan, in 2007, the Outstanding Youth Award from Taiwan Association of System Science and Engineering in 2012, the Dr. Wu, Da-You Memorial Research Award, National Science Council, Taiwan, in 2012, the Excellent Young Scholar Research Grant, National Science Council, Taiwan, in 2012-14, and the NTU Excellent Teaching Award in 2007, 2008, 2010-2013, 2018, and 2019. His current research interests include distributed and networked control systems, multiple dynamical agent systems, trajectory generation and path planning for autonomous vehicles.

Chien-Tung Chen received his MS degree in Computer Science from National ChengChi University in 2004. He is currently a software director in URSrobot Inc. His latest research interests include robot localization, robotic control and deep learning.

Chung-Hou Wu received his MS degree in Electrical Engineering from Chung Cheng Institute of Technology in 1997. He is currently the RD VP in URSrobot Inc. His latest research interests include satellite navigation, 5G communication, robot localization.



Chao-Cheng Chen received his MS degree in Electrical Engineering from National Taiwan University in 1992. He is currently the CEO in URSrobot Inc. His latest research interests include confluence of automatic robotics, robotics perception, robotic control.