Online Exploration and Coverage Planning in Unknown Obstacle-Cluttered Environments

Xinyue Kan, Hanzhe Teng, and Konstantinos Karydis

***Abstract*—Online coverage planning can be useful in applications like field monitoring and search and rescue. Without prior information of the environment, achieving resolution-complete coverage considering the non-holonomic mobility constraints in commonly-used vehicles (e.g., wheeled robots) remains a challenge. In this paper, we propose a hierarchical, hex-decomposition-based coverage planning algorithm for unknown, obstacle-cluttered environments. The proposed approach ensures resolution-complete coverage, can be tuned to achieve fast exploration, and plans smooth paths for Dubins vehicles to follow at constant velocity in real-time. Gazebo simulations and hardware experiments with a non-holonomic wheeled robot show that our approach can successfully tradeoff between coverage and exploration speed and can outperform existing online coverage algorithms in terms of total covered area or exploration speed according to how it is tuned.**

***Index Terms*—Nonholonomic Motion Planning, Robotics in Agriculture and Forestry, Online Coverage Planning.**

I. Introduction

THE paper addresses *online coverage planning* in unknown environments for vehicles with non-holonomic constraints. Efficient field coverage is essential for tasks such as environmental monitoring [1], map reconstruction [2], locating survivors [3], and autonomous exploration of forested areas [4]. In all these applications, regions to be explored may be unknown and partially observable. Even if the environment map can be obtained prior to departure, unexpected unvisitable areas may occur, such as collapsed trees following a storm. Hence, it is necessary to develop approaches that enable online exploration and coverage planning of irregularly-shaped environments with potential unexpected obstacles.

Depending on the application, various different types of unmanned vehicles–including aerial (fixed-wing aircraft), surface, ground (wheeled/legged robots), and underwater ones (e.g., [5]–[10])–can be deployed. A common challenge among most of them is the presence of non-holonomic mobility constraints, often manifested as a minimum turning curvature constraint. A way to take into consideration this constraint is by using a Dubins vehicle model [11], which specifies the vehicle to move in fixed-speed straight lines and counter/clockwise turns. More complex paths can be designed by concatenating straight-line and turning maneuvers.

When the environment is known, existing approaches (e.g., [6], [12]–[14]) decompose the region into a set of non-overlapping subregions, and then plan paths within each subregion. In practice, however, regions to be explored can be unknown. Exhaustive search strategies, like back-and-forth parallel swath motions [15], [16] or spiral paths [17], alter the robot's direction of motion if obstacles are encountered. This may lead to incomplete coverage when the region occluded by the obstacle has not been visited. In contrast, many existing online coverage methods [15], [18]–[20] lead to abrupt velocity and orientation changes when encountering obstacles. This effect becomes pronounced especially when operating in obstacle-cluttered environments, and can hinder tasks for which the success rate is sensitive to the quality of sensor input, e.g., in field reconstruction [2], [21] and survivor localization [22]. To mitigate this challenge, we propose an online approach that plans smooth trajectories that minimize the frequency of acceleration-deceleration events.

To represent the environment we consider a uniform hexagonal grid where a cell's dimension is determined by the robot's sensor footprint. Hexagon-based partitioning enjoys several benefits, including regular tessellation [23], uniform travel distance to all adjacent cells, and better description of non-convex regions [24]. The effectiveness of hexagonal cell decomposition has been shown in applications including potential-field-based path finding [25], field search [26] and offline path finding [27] in known environments, and online underwater mine countermeasure [28] with no restricted areas. Different from those approaches, we focus on describing **unknown**, **obstacle-cluttered**, bounded environments with duplicates of regular hexagons, which fill a plane with no gap or overlap.

We propose an online, hierarchical coverage planning approach for Dubins vehicles. At the high level, a Hex-Decomposition Coverage Planning (HDCP) algorithm is proposed. Based on information collected from the robot's observations up to the current time, the robot selects a feasible hexagon subregion to explore next. At the low level, Dubins-curve-based paths are planned in real-time. Closed-form solutions for feasible paths (e.g., start and goal positions for line segments, angles for arcs) are provided. The proposed HDCP algorithm aims to cover the entire unknown (yet bounded) environment, whereas its variant, HDCP-E, is used for fast exploration. The proposed method is evaluated in Gazebo

Manuscript received: February 24, 2020; Revised May 27, 2020; Accepted June 26, 2020.

This paper was recommended for publication by Editor Dezhen Song upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by NSF under grants #IIS-1724341 and #IIS-1901379, and ONR under grant #N00014-18-1-2252. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

X. Kan, H. Teng, and K. Karydis are with Dept. of Electrical and Computer Eng., Univ. of California, Riverside, 900 University Avenue, Riverside, CA 92521, USA. {xkan001, hteng007, karydis}@ucr.edu.

Digital Object Identifier (DOI): see top of this page.

simulations in three forest/farm-like environments and in one baseline empty environment, and compared against Spanning Tree Coverage (STC) [19], Boustrophedon motions and the A* search algorithm (BA*) [20], and Multi-robot Hex Decomposition Exploration (M-HDE) [29] in terms of covered area and exploration speed. We observe that unknown environments with random obstacles can be fully covered. HDCP covers the most amount of free space, whereas HDCP-E achieves the highest coverage area per unit time. The proposed method is evaluated experimentally with a non-holonomic wheeled robot.

Contributions: This paper has three key contributions.

1) We develop a hex-decomposition-based online coverage planning algorithm, HDCP, that guarantees resolution-completeness in unknown, cluttered spaces.
2) We propose the variant HDCP-E, to trade-off between fast exploration and resolution-completeness.
3) We offer closed-form solutions for planning smooth paths that robots can follow at constant speed.

The major difference between HDCP and our previous work M-HDE [29]–which also applies hexagonal cell decomposition in unknown environments–is the sensor-footprint-based decomposition strategy herein which ensures full coverage within each subregion. Further, M-HDE is developed mainly for online exploration tasks. If applied to coverage tasks (as in this work) it performs worse because it prioritizes visiting the most unexplored area. Doing so leaves uncovered subregions and necessitates returning back to fill in holes.

II. Related Work

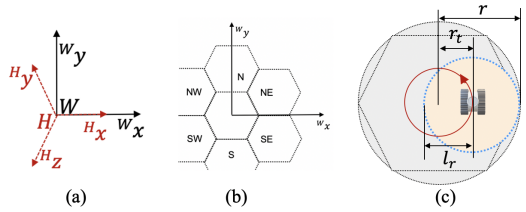
Several methods have been proposed to tackle the coverage path planning problem. When prior map information is available, planning can be offline [30]. Most planners use some form of decomposition, like Boustrophedon [6], [31], Semi-boustrophedon [14], Morse [12], or Line-sweep-based [13] decomposition, to partition the free space into a set of non-overlapping cells. For online coverage planning, using information collected by on-board sensors, similar cellular-decomposition-based strategies [15], [20], [32] are also applied. Resolution-complete coverage can be obtained by ensuring that all cells can be visited, and then applying "lawnmower" motions within each cell. Another popular approach used for coverage planning is the Spanning Tree algorithm [19], [33]. However, paths generated by those methods may contain sharp turns which can reduce efficiency and increase fuel consumption for non-holonomic robots [34].

A common way to consider non-holonomic constraints is to generate feasible paths using a Dubins model for offline coverage planning [6], [14], [35], [36]. Lewis et al. [14] solve the offline coverage problem as a traveling salesman problem, and add constraints to ensure planned paths consist of line segments and curves of a given minimum radius only. Yu et al. [37] proposed a graphical-optimization-based smooth planning strategy for Dubins vehicles. The method reduces the total coverage time, but at the expense of high computational complexity. Function-based smooth coverage planning methods generate paths represented by functions like clothoids [38] and Bézier curves [39], [40]. Due to their smoothness, Bézier curves enable fast coverage and energy efficiency, but at the price of complex calculations. Online coverage planning methods that are directly applicable to Dubins vehicles remain limited. One approach is to obtain an offline solution for coverage paths using any available prior knowledge, and then replan according to the information collected through sensors as the robot moves to avoid collisions [41]–[44]. Another way is to modify existing online coverage approaches, such as online STC [19] and online BA* [20], to decelerate and make smooth turns that satisfy the minimum turning radius constraints. Our proposed work fills in the gap by utilizing sensor-based decomposition and directly incorporating non-holonomic constraints.

III. Problem Setup

Consider a robot tasked to survey an unknown, bounded, obstacle-cluttered space $\mathcal{S}$. The robot is equipped with *navigation sensors* (e.g., LIDAR, depth camera) and *observation sensors* (e.g., RGB/thermal camera, mine detector). Navigation sensors are used to plan collision-free paths, while observation sensors are used to complete the designated task. Different from target search problems in which the search terminates once targets are located, the goal here is to cover the entire field with observation sensors.

We use two coordinate systems. Cartesian coordinates link to high-level objectives (e.g., to represent a point of interest in a map) and enable onboard sensor data inference (e.g., visual scene understanding). Cube coordinates are necessary to plan paths in hexagon subregions that form a hex grid. Thus, we use a two-layer environment map where a 2D hex grid plane is overlaid on top of a Cartesian plane.



(a) (b) (c)

Fig. 1. (a) Top view of the 2D hex frame H (dashed red) and world frame W (solid black). (b) Six adjacent cells of a subregion. (c) Covered area by the robot's observation sensors as it completes a circular path within a hex cell. The blue dashed circle filled with yellow depicts the sensor footprint, red solid lines represent paths, and the covered area is marked in gray.

We place a world (fixed) frame W (Fig. 1(a) solid black lines) in Cartesian plane, with axes ${}^W x$, ${}^W y$. The robot is modeled in Cartesian plane as a Dubins vehicle, i.e.

$$ {}^W\dot{x} = {}^W v \cos\theta \ , \ {}^W\dot{y} = {}^W v \sin\theta \ , \ \dot{\theta} = u_d \ , $$

where $({}^W x, {}^W y)$ is the robot's position, and θ is its heading. Speed v is constant, and $u_d \in \{-1, 0, 1\}$.

We place another a frame H (Fig. 1(a) dashed red lines) in hex grid plane as the frame of reference for cube coordinates. Cube coordinates correspond to three axes $({}^H x, {}^H y, {}^H z)$.[1] The directions of axes are given in Fig. 1(a).[2] The origin of frame H matches frame W, as well as a robot's departing position. Key variables used in this paper are listed in Table I.

[1]Cube coordinates have three axes in the 2D case.
[2]More details can be found in [29].

TABLE I
LIST OF KEY VARIABLES USED IN THE PAPER.

S_k, S_i, S_j	subregions
$({}^H x, {}^H y, {}^H z)$	cube coordinate in hex grid plane
$({}^W x, {}^W y)$	cartesian coordinate in Cartesian plane
r	hexagon grid radius
r_t	circular path radius
r_{min}	robot minimum turning radius
l_r	radius of observation sensor footprint
${}^W\mu_k$	starting point on circular path in S_k
${}^W\varphi_k$	tangent point on circular path in S_k
α	angle between ${}^W\mu_k$ and ${}^W\varphi_k$ in S_k
l	length of straight-line path for *Transitioning Mode*
τ	step index for hex grid plane planning
$\mathcal{P}_\tau$	a robot's path in hex grid plane
$\mathcal{V}_\tau$	a set of task-complete subregions
$\mathcal{E}_\tau$	a set of explored subregions by navigation sensor
$\Phi_{i,j}$, $\hat{\Phi}_{i,j}$	feasible path and shortest feasible path from S_i to S_j

In the hex grid plane, a hexagon subregion is S_k, with its cube coordinate $({}^H x_k, {}^H y_k, {}^H z_k)$. [3] The position of the center of S_k in frame W is

$$\begin{bmatrix} {}^W x \\ {}^W y \end{bmatrix} = \begin{bmatrix} \frac{3}{2}r & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2}r & -\frac{\sqrt{3}}{2}r \end{bmatrix} \begin{bmatrix} x_k & y_k & z_k \end{bmatrix}^T .$$

Each S_k has six adjacent cells (Fig. 1(b)), forming a set

$$\begin{aligned} \mathcal{N}(S_k) = \{&(x_k, y_k - 1, z_k + 1) \ , \ (x_k + 1, y_k - 1, z_k) \ , \\ &(x_k + 1, y_k, z_k - 1) \ , \ (x_k, y_k + 1, z_k - 1) \ , \\ &(x_k - 1, y_k + 1, z_k) \ , \ (x_k - 1, y_k, z_k + 1)\} \ . \end{aligned}$$

Hex side length r is determined based on the observation sensor footprint, taken here to be a circular disk of radius l_r (Fig. 1(c)).[4] Radius l_r depends on the selected sensor's range and is chosen by the user so that to achieve object detection of acceptable (by the user) accuracy. Once l_r is determined, the goal is to achieve full coverage for circumscribed circle of each hexagon cell. Suppose the radius of circular paths is r_t, chosen such as $r_{min} \leq r_t \leq l_r$. Ideally, setting $r_t = l_r$ ensures complete coverage of the circumscribed circle of a hexagon cell with no redundancy. However, in practice we may often have $r_t < l_r$ in which case more sensor data are collected but at the expense of efficiency (defined as newly covered area per unit time). Given l_r and r_t, then $r = l_r + r_t$.

Our method assumes the following. 1) The navigation sensor detection radius is large enough to at least detect obstacles in adjacent subregions. 2) There is sufficient battery life to achieve full coverage. 3) r_{min} is reasonably small compared to the sensor footprint and obstacle density.

IV. ONLINE HIERARCHICAL COVERAGE PLANNING

We propose a hierarchical approach to cover an unknown environment. High level planning runs on the hex grid. A robot determines a sequence of subregions to visit in the next several time steps (Section IV-A). Low level planning runs on the Cartesian plane. A robot plans circular and straight-line paths to complete an observation task in current subregion and move to next subregion, respectively (Section IV-B).

[3]We drop superscript H in cube coordinates for clarity of presentation.

[4]This is reasonable as there are a few ways to achieve a circular footprint, e.g., via rotation with gimbals or by using multiple sensors.

A. Hex Decomposition Coverage Planning

We first describe our proposed Hex Decomposition Coverage Planning (HDCP) approach. HDCP works at the hex grid plane, and ensures resolution-complete coverage of hex-decomposed unknown regions. In each subregion, a robot is deployed to complete the observation task, entering into an *Observing Mode*. A subregion is *visited*, if the robot has finished *Observing Mode* in this subregion. Re-entering a visited subregion will not trigger another *Observing Mode*. A subregion is *explored* if it has been covered by navigation sensors, i.e. having been marked as obstacle-free or obstacle-occupied region. Once a robot completes an observation task within one subregion, it enters into a *Transitioning Mode* and moves to another subregion. Only *Observing Mode* triggers observation sensors; navigation sensors collect information continuously during both modes.

We demonstrate the detailed process for HDCP in Algorithm 1. Upon departure, the robot initiates **robot-centric** frames H and W, whose origins are at robot's departure position, for high-level and low-level planning, respectively. A bounded unknown space $\mathcal{S}$ consists of unknown but finite number of subregions S_k. Let τ be the step of high-level planning which records when a subregion was visited, i.e. pair (τ, S_k) represents a robot's position in hex plane at step τ.

A robot's path in hex grid plane up to step τ is then defined as $\mathcal{P}_\tau = \{(t, S_k) | t \in [1, \tau], S_k \in \mathcal{S}\}$. Let $\mathcal{V}_\tau$ be the set containing visited subregions up to step τ, i.e. unique subregions in $\mathcal{P}_\tau$. $|\mathcal{V}_\tau| \leq |\mathcal{P}_\tau|$, where $|\cdot|$ denotes set cardinality. Let $\mathcal{E}_\tau = \{(S_k, u) | S_k \in \mathcal{S}, u \in \{0, 1\}\}$ be the set containing all explored subregions and their status u by navigation sensors up to step τ. $u = 0$ represents that a subregion is obstacle-free, otherwise $u = 1$. $\mathcal{E}_{\tau,u=0}$ returns all explored obstacle-free subregions, $\mathcal{E}_{\tau,u=1}$ returns obstacle-occupied, unvisitable subregions.

Suppose a robot finishes its observation task within subregion S_i at step τ, and needs to determine the next subregion S_j to visit at step $\tau + 1$ (line 10 of Algorithm 1). To decrease the number of repeatedly visited subregions, an unvisited subregion is preferred. To minimize the total travel distance, a robot prefers one of its adjacent subregions before moving to subregions further away (Fig. 2(a)). Let $\mathcal{C}_\tau = \mathcal{V}_\tau \cup \mathcal{E}_{\tau,u=1}$ be the set of all "undesired" choices of S_j, i.e. either already visited or obstacle-occupied. Then, set $\mathcal{Q}_\tau(S_i) = \{S_j^* \in \mathcal{N}(S_i) \mid S_j^* \notin \mathcal{C}_\tau\}, 0 \leq |\mathcal{Q}_\tau(S_i)| \leq 6$ contains candidates of S_j, denoted as S_j^*, which are unvisited, obstacle-free subregions adjacent to S_i.

As we seek to complete tasks for the entire free space within the unknown region efficiently, it is undesirable to leave any isolated subregion unvisited. The cost of coming back to "fill a hole" later can be avoided by finishing all nearby areas first before moving away. Let function $f(\cdot)$ calculate the number of visited or obstacle-occupied neighbors of a candidate S_j^* as $f(S_j^*) = |\mathcal{N}(S_j^*) \bigcap \mathcal{C}_\tau|$. $f(S_j^*) = 6$ indicates that all adjacent subregions of candidate S_j^* are either visited or obstacle-occupied, which makes this S_j^* a "hole" and hence should be prioritized to visit.

Algorithm 1 Hex Decomposition Coverage Planning

1: **procedure** HDCP
2: $\tau \leftarrow 1$, $S_i \leftarrow (0,0,0)$, empty sets $\mathcal{P}_\tau$, $\mathcal{V}_\tau$, $\mathcal{E}_\tau$, $\hat{\Phi}_{i,j}$
3: **while** $S_i \neq \emptyset$ **do**
4: Move to S_i according to Eq. (7) and Appendix
5: $\mathcal{P}_\tau \leftarrow \mathcal{P}_\tau \cup (\tau, S_i)$
6: **if** $S_i \notin \mathcal{V}_\tau$ **then**
7: Observing mode, $\mathcal{V}_\tau \leftarrow \mathcal{V}_\tau \cup S_i$
8: **end if**
9: Update $\mathcal{E}_\tau$ by navigation sensor results
10: $S_i, \hat{\Phi}_{i,j} \leftarrow \text{NextHex}(S_i, \mathcal{E}_\tau, \hat{\Phi}_{i,j})$
11: $\tau \leftarrow \tau + 1$
12: **end while**
13: **end procedure**
14: **procedure** NEXTHEX(S_i, $\mathcal{E}_\tau$, $\hat{\Phi}_{i,j}$)
15: **if** $\mathcal{Q}_\tau(S_i) \neq \emptyset$ **then**
16: Obtain S_j according to Eq. (1), $\hat{\Phi}_{i,j} \leftarrow \emptyset$
17: **else if** $\mathcal{Q}'_\tau \neq \emptyset$ **then**
18: **if** $|\mathcal{E}_\tau| \neq |\mathcal{E}_{\tau-1}|$ **or** $\hat{\Phi}_{i,j} = \emptyset$ **then**
19: Obtain S_j according to Eq. (2)
20: $\hat{\Phi}_{i,j} \leftarrow \text{GetAStarPath}(S_i, S_j, \mathcal{E}_\tau)$
21: **else** $S_j \leftarrow \hat{\Phi}_{i,j}[1]$, $\hat{\Phi}_{i,j} \leftarrow \hat{\Phi}_{i,j}[2:end]$
22: **end if**
23: **else** $S_j \leftarrow \emptyset$, $\hat{\Phi}_{i,j} \leftarrow \emptyset$
24: **end if**
25: **return** S_j, $\hat{\Phi}_{i,j}$
26: **end procedure**

When $|\mathcal{Q}_\tau(S_i)| > 0$, there exists at least one candidate S_j^* that is adjacent to S_i. Under this condition, among all candidates S_j^*, the next subregion that a robot prioritizes to visit, S_j, is determined as

$$\begin{aligned} \arg\max_{S_j^*} \quad & f(S_j^*) \\ \text{s.t.} \quad & S_j^* \in \mathcal{Q}_\tau(S_i) \ , \ |\mathcal{Q}_\tau(S_i)| > 0 \ . \end{aligned} \tag{1}$$

Given (1), among all unvisited adjacent subregions the robot will choose the one with the maximum visited/obstacle-occupied neighbors as S_j (line 16 of Algorithm 1). This strategy ensures that the robot does not leave any isolated unvisited subregions, to avoid the need to return to this area.

If $|\mathcal{Q}_\tau(S_i)| = 0$, the task has been completed for all adjacent subregions of S_i. Under this condition, the robot will move to a subregion S_j that is nonadjacent to S_i (Fig. 2(b)). Let $\mathcal{Q}'_\tau = \{S_j^* \in \mathcal{V}_\tau \mid \mathcal{N}(S_j^*) \setminus \mathcal{C}_\tau \neq \emptyset\}$ be the set that contains all candidates S_j^*, which are visited subregions with unvisited obstacle-free neighbors. If $\mathcal{Q}'_\tau$ is not empty, the robot selects S_j from all candidates S_j^* as

$$\begin{aligned} \arg\max_{S_j^*} \quad & t \\ \text{s.t.} \quad & (t, S_j^*) \in \mathcal{P}_\tau \ , \ S_j^* \in \mathcal{Q}'_\tau \ , \ t \in [1, \tau] \ . \end{aligned} \tag{2}$$

Per (2), the robot revisits a nearest-in-the-past visited subregion S_j which has unvisited obstacle-free adjacent subregions (line 19 of Algorithm 1). Once S_j is chosen, we seek a path in hex grid, which contains a sequence of subregions on the way to move to S_j. In order to determine an optimal feasible path, we need some definitions.

Definition 1: A *feasible path* between obstacle-free subregions S_i and S_j in hex grid plane, denoted as $\Phi_{i,j}$, is a list of ordered obstacle-free subregions[5] such that if only the movement to an adjacent subregion is allowed for each step, a robot starting from S_i can reach S_j in finite steps.

Definition 2: A *shortest feasible path*, denoted as $\hat{\Phi}_{i,j}$, is the path among all feasible paths $\Phi_{i,j}$ which contains the least number of subregions.

Definition 3: A subregion is *visitable* if and only if 1) this subregion is obstacle-free, and 2) there exists one (shortest) feasible path from the departure position to this subregion.

Each subregion in the *feasible path* and *shortest feasible path* is an adjacent subregion of its preceding and following subregions in the ordered list. The *feasible path* and *shortest feasible path* are not unique between two subregions. From the current position S_i, there must exist at least one *feasible path* to any subregion on path $\mathcal{P}_\tau$, which is a subset of path $\mathcal{P}_\tau$.

A *shortest feasible path* $\hat{\Phi}_{i,j}$ can be obtained according to function *GetAStarPath* (line 20 of Algorithm 1). Given all explored subregions $\mathcal{E}_\tau$, *GetAStarPath* applies the A^* algorithm [45] with distance cost q in 2D hex grid

$$q(S_k, S_{k'}) = (|x_k - x_{k'}| + |y_k - y_{k'}| + |z_k - z_{k'}|)/2 \ . \tag{3}$$

While following $\hat{\Phi}_{i,j}$ (line 21 of Algorithm 1), more subregions are explored by the navigation sensor. If $\mathcal{E}_\tau$ is updated, $\hat{\Phi}_{i,j}$ needs to be updated accordingly (lines 18-20 of Algorithm 1). Note that $\hat{\Phi}_{i,j}$ may contain 1) unvisited subregions, and 2) subregions with unvisited obstacles-free neighbors. If at any step, an unvisited subregion is encountered, the robot enters *Observing Mode* to complete the task for the subregion (lines 6-8 of Algorithm 1), then switches to *Transitioning Mode*. If a subregion with unvisited obstacle-free neighbors is encountered, we discard $\hat{\Phi}_{i,j}$ and move to an adjacent unvisited subregion (line 16 of Algorithm 1).

When $\mathcal{Q}_\tau(S_i) = \mathcal{Q}'_\tau = \emptyset$, coverage is complete. HDCP establishes that in a bounded environment, for given discretization resolution, the coverage process terminates in finite time, leaving no unvisited area that is visitable according to Definition 3. Coverage using HDCP is complete to the resolution of the smallest allowed hexagon cell, which can be referred as resolution-completeness [46].

Lemma 1 *In an unknown, bounded environment, online coverage using HDCP (Algorithm 1) is resolution-complete.*

Proof of Lemma 1 We prove Lemma 1 by contradiction. Assume the coverage terminates according to Algorithm 1 and there still exists a *visitable*, yet unvisited, subregion S_q. Suppose the departure position of the coverage is S_p. Since S_q is *visitable*, there must exist a *shortest feasible path* $\hat{\Phi}_{p,q}$ according to Definition 3. Denote the mth subregion on the *shortest feasible path* as $\hat{\Phi}_{p,q}[m]$.

According to Definition 1, $\hat{\Phi}_{p,q}[1]$ must be an adjacent subregion to S_p, $\hat{\Phi}_{p,q}[1] \in \mathcal{N}(S_p)$. Since the progress has terminated, $\mathcal{Q}'_\tau = \emptyset$. Recalling the definition of $\mathcal{Q}'_\tau$, we can deduce (6) from (4) and (5),

[5]The first subregion in the ordered list is adjacent to S_i, while the last subregion is S_j itself.

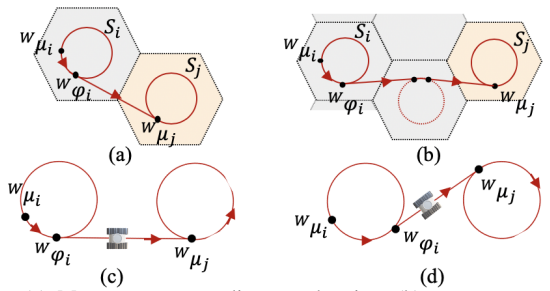


Fig. 2. (a) Movement to an adjacent subregion, (b) movement to a non-adjacent subregion, (c) outer tangent line path, (d) inner tangent line path.

$$\hat{\Phi}_{p,q}[1] \in \mathcal{N}(S_p) \ , \tag{4}$$

$$\mathcal{Q}'_\tau = \emptyset \Rightarrow \mathcal{N}(S_p) \subset \mathcal{C}_\tau \ , \tag{5}$$

$$\hat{\Phi}_{p,q}[1] \in \mathcal{N}(S_p) \ , \ \mathcal{N}(S_p) \subset \mathcal{C}_\tau \Rightarrow \hat{\Phi}_{p,q}[1] \in \mathcal{C}_\tau \ . \tag{6}$$

Since $\hat{\Phi}_{p,q}[1] \notin \mathcal{E}_{\tau,u=1}$, then $\hat{\Phi}_{p,q}[1] \in \mathcal{V}_\tau$. Similarly, we have $\hat{\Phi}_{p,q}[m] \in \mathcal{V}_\tau$ for $m = 2, \cdots, |\hat{\Phi}_{p,q}|$. Therefore, $S_q = \hat{\Phi}_{p,q}[|\hat{\Phi}_{p,q}|] \in \mathcal{V}_\tau$. Hence, S_q is already visited, which leads to the contradiction.

B. Dubins Path Planning

Next, we discuss Dubins Path Planning for *Observing Mode* and *Transitioning Mode*. For a robot currently in S_i, it first enters into *Observing Mode*. For an observation task, it is required that the observation sensor covers the entire subregion, which leads to a circular path. Let ${}^W\mu_i$ be the starting point on the circular path. The robot follows a full circle, $(C_{2\pi})_i$, and then returns to ${}^W\mu_i$. Once arriving at ${}^W\mu_i$, the robot enters into *Transitioning Mode*, in which the robot aims to move to the starting point ${}^W\mu_j$ for the next subregion S_j. Constrained by the vehicle model, feasible paths always consist of arcs and straight lines, without turns sharper than robot's minimum turning capability.

Figures 2(c), (d) show the planned path, which comprises an arc of angle α and a straight line of length l. The straight line is chosen to be tangent to both circular paths in S_i and S_j. Let the tangent point in S_i be ${}^W\varphi_i$, the corresponding tangent point in S_j will become the starting point ${}^W\mu_j$. Among outer tangent points (Fig. 2(c)) and inner tangent points (Fig. 2(d)), following the robot's current moving direction, we select the one that is the closest to ${}^W\mu_i$ along the circular path. Using geometry (closed-form solutions for tangent points are given in the Appendix), α and l are

$$\begin{aligned} \alpha &= \cos^{-1}(1 - \|{}^W\mu_i - {}^W\varphi_i\|^2/2r_t^2) \ , \\ l &= \|{}^W\mu_i - {}^W\varphi_i\| \ . \end{aligned} \tag{7}$$

The combined path for *Observing* and *Transitioning Mode* between S_i and S_j is $(C_{2\pi})_i(C_\alpha L_l)_{i\to j}$. By utilizing tangent points to switch among arcs and straight line paths, generated combined paths are smooth.

C. Trading-off Exploration Coverage and Speed

In some scenarios such as waypoint coverage [9] and water sample collection [47], full coverage within subregions does not need to be enforced. In this case, the exploration process of unknown environments can be accelerated by combining *Observing* and *Transitioning Mode*, which is referred to as HDCP-E. In HDCP-E, full circle trajectory $(C_{2\pi})_i$ is removed from planned paths to accelerate exploration of more hexagon subregions. At the same time, observation sensors are enabled throughout the process. The modified paths become $(C_\alpha L_l)_{i\to j}$. Hence, HDCP-E inherits the advantage of smooth and continuous paths from HDCP.

The two variants (HDCP and HDCP-E) reveal the trade-off between exploitation and exploration. HDCP guarantees resolution-complete coverage of unknown yet bounded space by 1) visiting all subregions, and 2) achieving full coverage within each subregion, whereas HDCP-E enables fast and complete exploration in terms of visiting all subregions. Importantly, the next-subregion-selection strategy (Algorithm 1) can also be applied to other cell-decomposition-based coverage algorithms to obtain the order of visiting subregions.

V. Experiments, Results, and Discussion

Our proposed variants are evaluated in Gazebo simulation and experimentally with a non-holonomic wheeled robot. Their performance is compared against 1) online BA* [20], 2) online STC [19], and 3) M-HDE [29] in terms of total coverage area and average exploration speed.

A. Simulation

1) Simulation Setup: Figures 3(a)-(c) show three 20m×20m 2D simulated forest/farm-like environments in Gazebo (random, uniform, and in-row placement, respectively). Two type of trees, which are different in terms of size and shape, are used. Ten trees of each type are placed in each environment, hence all three environments have same amount of free space. In the random environment (Fig. 3(a)), trees are placed randomly to represent a forest-like unstructured environment. In the uniform environment (Fig. 3(b)), trees are arranged and lined up strictly. In the in-row environment (Fig. 3(c)), trees are loosely lined up, with slightly-varying spacing between them. This environment approximates a more realistic intercropping agricultural field. We further consider the baseline scenario of operating in an empty environment bounded by square walls.

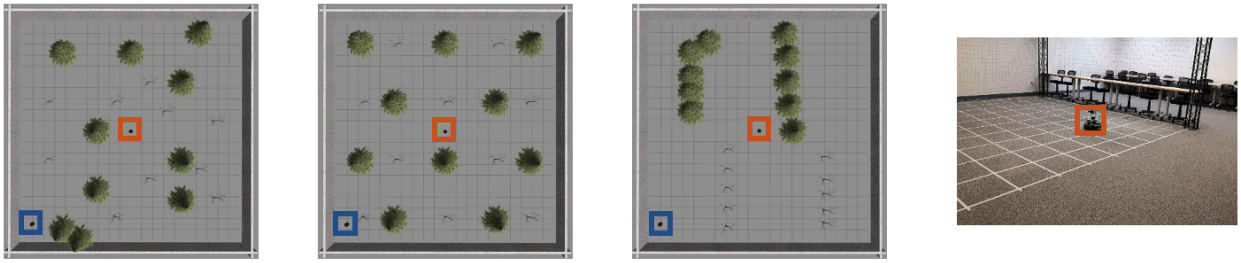
We deploy the non-holonomic wheeled robot Turtlebot to cover the entire area without prior knowledge of the environment map. The robot is equipped with an RPLidar laser scanner as the navigation sensor, and an Astra Pro stereo camera as the observation sensor. Both perception[6] and planning are online. To reduce the uncertainty in obstacle detection caused by online perception and odometry drift, we run ten trials for every obstacle-cluttered scenario.

Camera observations are chosen to have high accuracy with $l_r = 0.5$ m. Hence, for HDCP, HDCP-E, and M-HDE, the hex subregion side length is $r = 1$ m. The robot moves at a constant velocity of 1 m/s. For BA* and STC, the square side length is 1 m;[7] the robot moves forward at 1 m/s, and decelerates to 0.3 m/s when taking sharp turns.[8]

[6]We use the open-source LIDAR-based obstacle detector published in https://github.com/tysik/obstacle_detector.

[7]We refer the reader to [19], [20] for more details.

[8]We optimized the turning speed so that no odometry drift is observed.



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

Fig. 3. In simulation, (a) random, (b) uniform, and (c) in-row environments. (d) Experimental environment.

TABLE II
RESULTS FOR PERCENTAGE OF COVERED AREA, RUNTIME AND EXPLORATION SPEED OVER 10 SIMULATION TRIALS

depart at center	**random**			**uniform**			**in-row**		
	area (%)	runtime (s)	avg (m^2/s)	area (%)	runtime (s)	avg (m^2/s)	area (%)	runtime (s)	avg (m^2/s)
HDCP	**83.2**±0.4	701.2±12.6	0.47	**80.3**±0.7	689.7±10.3	0.46	**79.1**±1.1	700.3±11.7	0.44
HDCP-E	68.1±0.4	337.3±8.0	**0.79**	68.3±0.6	342.7±4.4	**0.78**	66.8±0.8	352.4±11.3	**0.75**
STC	52.7±1.4	561.7±12.1	0.37	58.6±0.1	627.6±2.6	0.37	13.5±0.1	144.2±0.4	0.37
M-HDE	70.4±1.3	450.0±10.6	0.62	71.8±1.7	485.1±15.4	0.58	68.3±2.4	463.6±12.2	0.58
BA*	75.7±1.1	629.4±18.5	0.46	73.6±0.5	579.5±4.5	0.48	75.1±1.7	635.1±7.3	0.45
depart at lower-left	**random**			**uniform**			**in-row**		
	area (%)	runtime (s)	avg (m^2/s)	area (%)	runtime (s)	avg (m^2/s)	area (%)	runtime (s)	avg (m^2/s)
HDCP	**83.2**±0.4	727.6±6.2	0.45	**85.7**±1.3	740.4±11.3	0.46	**81.3**±1.4	699.2±20.6	0.45
HDCP-E	70.7±1.1	369.3±10.0	**0.75**	67.8±1.1	338.9±9.2	**0.79**	69.2±1.5	357.3±10.9	**0.76**
STC	65.8±3.7	622.8±19.1	0.42	59.8±1.9	536.5±29.6	0.44	61.0±1.6	502.4±26.3	0.48
M-HDE	74.0±1.6	496.9±10.6	0.59	75.5±1.8	510.8±22.6	0.58	69.8±1.9	468.3±20.4	0.59
BA*	80.7±0.7	661.1±21.2	0.46	79.0±1.3	522.8±30.0	0.57	79.2±4.3	666.2±52.5	0.47

To eliminate the influence of starting position, the robot is deployed from 2 different positions: center and lower-left corner (red and blue squares in Fig. 3(a)-(c)). Note that in both simulation and real experiment, robot-centric hex (for HDCP, HDCP-E, and M-HDE) and square (for STC and BA*) grids originate at robot's departure position. In a robot-centric grid, the number of occupied cells caused by obstacles and environment boundaries is influenced by the relative position between the object and the origin of the grid. For instance, a small obstacle can either lie within one cell, or on an edge/intersection of multiple cells after varying the grid origin. In addition, the size of cells is determined according to the robot footprint. The occupied cells near boundaries are marked as obstacles by the robot at runtime via onboard perception. We do not pre-determine near-boundary inaccessible cells for the robot prior to departure.

2) Results and Discussion: Table II contains means and one-standard deviations for the percentage of coverage area over total free space, total algorithm runtime, and averaged exploration speed over ten trials. We consider the total free space as subtracting the tree-occupied area from the total environment area, which is consistent among all scenarios.

Results suggest that, regardless of departing position, HDCP covers the most area in all evaluated, obstacle-cluttered environments. While the uniform and in-row environments bounded by square walls are more regular and structured, HDCP still outperforms the other evaluated methods that use a square grid discretization. Our findings demonstrate the advantage of hexagon decomposition, where we mark obstacles as more "round-like" hexagons instead of square cells in obstacle-cluttered environments.

In terms of exploration speed, HDCP-E covers almost twice as fast as STC and BA* in all environments from both

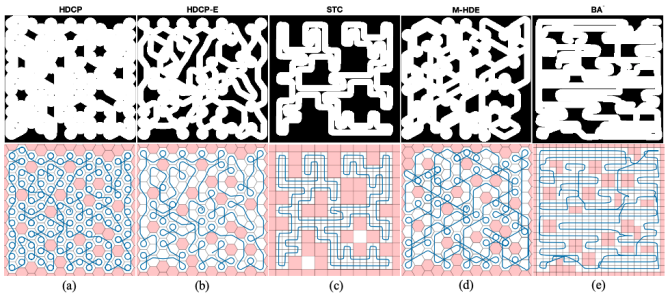


Fig. 4. Covered area (top panels, in white), detected obstacles (bottom panels, red cells), and corresponding robot paths (blue curves) for (a) HDCP, (b) HDCP-E, (c) M-HDE, (d) STC, and (e) BA* in the random map when the robot departs at center position.

departure positions. The results suggest that the strategy of HDCP and HDCP-E for selecting the next subregion is efficient in terms of exploring more unknown space. However, in HDCP, more area is covered when following the full-circle at the expense of exploration speed. M-HDE from our previous work [29] achieves the second fastest exploration, in which the lack of inner-tangent straight-line transitioning causes longer paths.[9] Hence, in scenarios when the speed of exploring more unknown space is the main concern, HDCP-E can be used to achieve fast exploration.

It is worth noticing that STC is the approach most sensitive to the environment and departure position. Especially in the in-row environment, STC is unable to cover the entire space when departing from center. This is because STC assumes all visitable space has to have a width of at least four times of the sensor footprint radius, to ensure repetition-free paths [19].

[9]Note that we modify M-HDE by replacing the original path finding strategy in [29] with A* to achieve better performance.

If the assumption is not satisfied, the area will be marked as obstacle, even if the width of the area is wider than the robot's own width and can thus be visited. On the other hand, for HDCP, changing departure position and environment can cause at most 6% coverage percentage loss.

Evident in Fig. 4, our proposed method generates smooth paths for robots with non-holonomic constraints. The robot moves at constant speed throughout the process, which enables better path following and sensor stability. In contrast, small areas can remain uncovered in both BA$*$ and STC when the robot fails to follow planned paths exactly due to abrupt deceleration before turning. The advantage of smooth paths is more obvious in cluttered environments, in which case turning maneuvers are required more frequently.

We also evaluate the algorithm in an square environment without obstacles, which is believed to be most suitable for lawnmower-like methods such as BA* and STC. In the empty environment, BA* ($0.75\ m^2/s$) achieves comparable coverage speed as HDCP-E ($0.82\ m^2/s$). This is because when applying lawnmower-like methods in the empty environment, the robot follows straight-line paths from one side to another, which requires minimum number of turns. In addition, BA* (92.1%) covers slightly more area than HDCP (91.4%). This is because square cells describe the square space better when no obstacle exists. However, in obstacle-cluttered environments, the existence of unexpected obstacles force the robots to take frequent, possibly sharp turns. Taking sharp turns requires deceleration and acceleration for lawnmower-like methods, whereas HDCP and HDCP-E allow robots to operate at constant velocity. Moreover, as discussed above, the advantage of describing obstacles with hex cells is more obvious in obstacle-cluttered environments than empty environments. Overall, the proposed HDCP and HDCP-E work well in unknown and irregularly-shaped obstacle-cluttered environments that are bounded.

B. Experiments

We also evaluate the performance of all algorithms in a $10\text{m} \times 8\text{m}$ indoor space (Fig. 3(d)) with a real Turtlebot robot configured as in the simulation. The environment contains a truss, desks, and chairs as obstacles. For HDCP, HDCP-E, and M-HDE, we have $l_r = 0.4$ m and $r = 0.8$ m; the robot moves at constant velocity of 0.3 m/s. For BA* and STC, we have side length of 0.8 m, and the robot moves forward at 0.3 m/s, and decelerates to 0.1 m/s in turning.

TABLE III
PERCENTAGE OF COVERED AREA, RUNTIME AND EXPLORATION SPEED IN EXPERIMENTS

	HDCP	HDCP-E	STC	M-HDE	BA*
area (%)	**76.5**	62.3	65.6	73.5	75.8
runtime (s)	367.0	187.5	268.5	259.0	306.5
avg (m^2/s)	0.10	**0.16**	0.12	0.14	0.12

We observe that HDCP and BA* cover similar area, which is expected since the experimental environment is mostly empty with obstacles lined up strictly in the middle. HDCP-E has the highest average exploration speed, 0.16 m^2/s, which is consistent to simulation results. Further, our approach generates smooth paths (Fig. 5) in real time for robots to follow at constant speed, while BA* and STC both require frequent acceleration and deceleration since the robot makes turns frequently in the small, cluttered space.

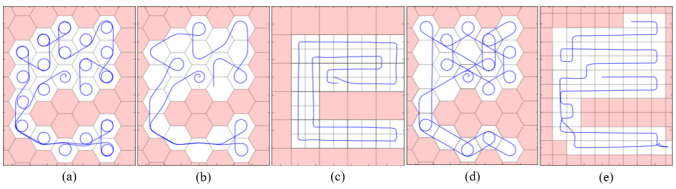


Fig. 5. Paths for (a) HDCP, (b) HDCP-E, (c) STC, (d) M-HDE, and (e) BA* in hardware experiments.

VI. CONCLUSIONS

The paper contributes to *online resolution-complete coverage planning* in unknown obstacle-cluttered environments. Research on this vein is limited when it comes to considering some form of dynamic feasibility (in this case Dubins vehicles), while ensuring resolution-complete coverage.

Results suggest that our proposed algorithm HDCP can cover more area compared to existing methods such as M-HDE, STC and BA* in unstructured and obstacle-cluttered environments, due to the advantage of decomposing the workspace in hex cells. Its variant HDCP-E achieves the fastest exploration (covered area per unit time) in both structured and unstructured environments. Further, we show that Dubins vehicles may fail to follow frequently required sharp turns using STC and BA*, leading to uncovered areas along the search path. Our method guarantees resolution-complete coverage while considering non-holonomic constraints in the form of Dubins curves. Derived geometric closed-form solutions to determine how to move between subregions enable real-time planning. Current weaknesses of HDCP are a slight underperformance compared to BA* in square, empty environments, and the presence of repeated arc segments on circular paths when moving to the tangent points. However, we believe that ensuring path smoothness and continuity due to this repetition outweighs the limitation.

Future work will focus on performing a complexity analysis to investigate how to speed up the methodology, and application to real world agricultural/forest environments.

APPENDIX

Suppose the center position of S_k, $S_{k'}$ in the Cartesian plane are (a_k, b_k), $(a_{k'}, b_{k'})$, respectively. $w = (a_{k'} - a_k)^2 + (b_{k'} - b_k)^2$. Inner tangent points are

$$
\begin{aligned}
{}^{W}\varphi_k(x) &= \frac{2r_t^2(a_{k'} - a_k) \pm r_t(b_{k'} - b_k)\sqrt{w - (2r_t)^2}}{w} + a_k \ , \\
{}^{W}\varphi_k(y) &= \frac{2r_t^2(b_{k'} - b_k) \pm r_t(a_k - a_{k'})\sqrt{w - (2r_t)^2}}{w} + b_k \ , \\
{}^{W}\mu_{k'}(x) &= \frac{2r_t^2(a_k - a_{k'}) \pm r_t(b_k - b_{k'})\sqrt{w - (2r_t)^2}}{w} + a_{k'} \ , \\
{}^{W}\mu_{k'}(y) &= \frac{2r_t^2(b_k - b_{k'}) \pm r_t(a_{k'} - a_k)\sqrt{w - (2r_t)^2}}{w} + b_{k'} \ .
\end{aligned}
$$

Outer tangent points are

$$
{}^{W}\varphi_k(x) = a_k \pm r_t\frac{(b_k - b_{k'})}{\sqrt{w}} \ , \ {}^{W}\varphi_k(y) = b_k \pm r_t\frac{(a_{k'} - a_k)}{\sqrt{w}} \ ,
$$

$$
{}^{W}\mu_{k'}(x) = a_{k'} \pm r_t\frac{(b_k - b_{k'})}{\sqrt{w}} \ , \ {}^{W}\mu_{k'}(y) = b_{k'} \pm r_t\frac{(a_{k'} - a_k)}{\sqrt{w}} \ .
$$

REFERENCES

[1] M. Popović, T. Vidal-Calleja, G. Hitz, I. Sa, R. Siegwart, and J. Nieto, "Multiresolution mapping and informative path planning for UAV-based terrain monitoring," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, 2017, pp. 1382–1388.

[2] J. Dong, J. G. Burnham, B. Boots, G. Rains, and F. Dellaert, "4D crop monitoring: Spatio-temporal reconstruction for agriculture," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2017, pp. 3878–3885.

[3] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective UAV path planning for search and rescue," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2017, pp. 5569–5574.

[4] A. Jaakkola, J. Hyyppä, X. Yu, A. Kukko, H. Kaartinen, X. Liang, H. Hyyppä, and Y. Wang, "Autonomous collection of forest field reference—the outlook and a first step with UAV laser scanning," *Remote Sensing*, vol. 9, no. 8, 2017.

[5] T. Shima, S. Rasmussen, and D. Gross, "Assigning micro UAVs to task tours in an urban terrain," *IEEE Trans. on Control Syst. Technology*, vol. 15, no. 4, pp. 601–612, 2007.

[6] N. Karapetyan, J. Moulton, J. S. Lewis, A. Quattrini Li, J. M. Okane, and I. Rekleitis, "Multi-robot dubins coverage with autonomous surface vehicles," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2018, pp. 2373–2379.

[7] P. Tokekar, N. Karnad, and V. Isler, "Energy-optimal trajectory planning for car-like robots," *Autonomous Robots*, vol. 37, no. 3, pp. 279–300, Oct 2014.

[8] Bingxi Li, B. Moridian, and N. Mahmoudian, "Underwater multi-robot persistent area coverage mission planning," in *OCEANS MTS/IEEE Monterey*, 2016, pp. 1–6.

[9] N. Karapetyan, J. Moulton, and I. Rekleitis, "Dynamic autonomous surface vehicle control and applications in environmental monitoring," in *OCEANS MTS/IEEE SEATTLE*, 2019, pp. 1–6.

[10] S. Rahman, A. Q. Li, and I. Rekleitis, "Sonar visual inertial slam of underwater structures," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2018, pp. 5190–5196.

[11] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American J. of Maths.*, vol. 79, pp. 497–516, 1957.

[12] E. Galceran and M. Carreras, "Efficient seabed coverage path planning for asvs and auvs," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2012, pp. 88–93.

[13] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2001, pp. 27–32.

[14] J. S. Lewis, W. Edwards, K. Benson, I. Rekleitis, and J. M. O'Kane, "Semi-boustrophedon coverage with a dubins vehicle," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Syst. (IROS)*, 2017, pp. 5630–5637.

[15] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions," *The Int. J. of Robotics Research*, vol. 21, no. 4, pp. 345–366, 2002.

[16] A. Zelinsky, R. Jarvis, J. C. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Int. Conf. on Advanced Robotics*, 1993, pp. 533–538.

[17] M. Bosse, N. Nourani-Vatani, and J. Roberts, "Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2007, pp. 698–703.

[18] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Syst.*, vol. 59, no. 10, pp. 801 – 812, 2011.

[19] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Annals of Maths. and Artificial Intell.*, vol. 31, no. 1, pp. 77–98, Oct 2001.

[20] H. Viet, V.-H. Dang, M. Laskar, and T. Chung, "Ba: An online complete coverage algorithm for cleaning robots," *Applied Intell.*, vol. 39, 2013.

[21] Y. Chen, S. Huang, and R. Fitch, "Active slam for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Trans. on Mechatronics*, pp. 1–1, 2019.

[22] N. Doulamis, P. Agrafiotis, G. Athanasiou, and A. Amditis, "Human object detection using very low resolution thermal cameras for urban search and rescue," in *Int. Conf. on PErvasive Tech. Related to Assistive Env.*, 2017.

[23] J. Gullberg, P. Hilton, and P. Gullberg, *Mathematics: From the Birth of Numbers*, 1997.

[24] Z. A. Algfoor, M. S. Sunar, and H. Kolivand, "A comprehensive study on pathfinding techniques for robotics and video games," *Int. J. of Computer Games Technology*, vol. 2015, p. 7, 2015.

[25] E. S. H. Hou and D. Zheng, "Hierarchical path planning with hexagonal decomposition," in *IEEE Int. Conf. on Syst., Man, and Cybernetics*, 1991, pp. 1005–1010.

[26] H. Azpúrua, G. M. Freitas, D. G. Macharet, and M. F. M. Campos, "Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys," *Robotica*, vol. 36, no. 8, pp. 1144–1166, 2018.

[27] J. Yu and D. Rus, "An effective algorithmic framework for near optimal multi-robot path planning," in *Int. J. of Robotics Research*, 2018, pp. 495–511.

[28] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. on Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2013.

[29] X. Kan, H. Teng, and K. Karydis, "Multi-robot field exploration in hex-decomposed environments for dubins vehicles," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2019, pp. 449–455.

[30] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of Math. and artificial intell.*, vol. 31, no. 1-4, pp. 113–126, 2001.

[31] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2010, pp. 5525–5530.

[32] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards robust on-line multi-robot coverage," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2006, pp. 1710–1715.

[33] N. Agmon, N. Hazon, and G. A. Kaminka, "Constructing spanning trees for efficient multi-robot coverage," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2006, pp. 1698–1703.

[34] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges," *J. Inf. Sci. Eng.*, vol. 33, pp. 101–121, 2017.

[35] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, Apr 2014.

[36] K. Savla, F. Bullo, and E. Frazzoli, "The coverage problem for loitering dubins vehicles," in *IEEE Conf. on Decision and Control*, 2007, pp. 1398–1403.

[37] X. Yu, T. A. Roppel, and J. Y. Hung, "An optimization approach for planning robotic field coverage," in *Conf. of the IEEE Industrial Electronics Society (IECON)*, 2015, pp. 4032–4039.

[38] D. Sabelhaus, F. Röben, L. P. M. zu Helligen, and P. S. Lammers, "Using continuous-curvature paths to generate feasible headland turn manoeuvres," *Biosyst. Engr.*, vol. 116, no. 4, pp. 399 – 409, 2013.

[39] A. Khan, I. Noreen, and Z. Habib, "Coverage path planning of mobile robots using rational quadratic bézier spline," in *Int. Conf. on Frontiers of Information Technology (FIT)*, 2016, pp. 319–323.

[40] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Syst.*, vol. 59, no. 10, pp. 801 – 812, 2011.

[41] M. Carreras, J. D. Hernández, E. Vidal, N. Palomeras, and P. Ridao, "Online motion planning for underwater inspection," in *IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2016, pp. 336–341.

[42] R. Pěnička, M. Saska, C. Reymann, and S. Lacroix, "Reactive dubins traveling salesman problem for replanning of information gathering by UAVs," in *European Conf. on Mobile Robots*, 2017, pp. 1–6.

[43] J. Marek, P. Váňa, and J. Faigl, "Trajectory planning for aerial vehicles in the area coverage problem with nearby obstacles," in *Modelling and Simulation for Autonomous Syst.*, 2019, pp. 226–236.

[44] Y. Guo and M. Balakrishnan, "Complete coverage control for nonholonomic mobile robots in dynamic environments," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2006, pp. 1704–1709.

[45] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Syst. Sci. and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[46] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[47] S. Manjanna, A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek, "Heterogeneous multi-robot system for exploration and strategic water sampling," in *IEEE Int. Conf. on Robotics and Autom. (ICRA)*, 2018, pp. 4873–4880.