IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 16 August 2023, accepted 20 August 2023, date of publication 24 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3308145

RESEARCH ARTICLE

Full Coverage of Confined Irregular Polygon Area for Marine Survey

JI-HONG LI, (Senior Member, IEEE), HYUNGJOO KANG, MIN-GYU KIM, HANSOL JIN, MUN-JIK LEE, GUN RAE CHO, (Member, IEEE), AND CHULHEE BAE
Autonomous Systems Research and Development Division, Korea Institute of Robotics and Technology Convergence, Pohang 37666, Republic of Korea

Corresponding author: Ji-Hong Li (jhli5@kiro.re.kr)

This work was supported in part by the Project titled ''Autonomous Underwater Vehicle Fleet and its Operation System Development for Quick Response of Search on Maritime Disasters'' of the Korea Institute of Marine Science and Technology Promotion (KIMST) funded by the Korea Coast Guard Agency under Grant KIMST-20210547, and in part by KIMST funded by the Ministry of Oceans and Fisheries in Republic of Korea under Grant RS-2023-00256122.

ABSTRACT This paper considers the coverage path planning (CPP) problem for marine surveys, where the survey vehicles' movements are confined in the irregular polygon search area. So this is a sort of milling problem with the minimum number of turns in order for improvement of the acquired sonar image quality. For this purpose, this paper proposes a novel method called CbSPSA (calculation based shortest path search algorithm). Especially in the irregular corner area, this algorithm can easily calculate a kind of shortest path for its full coverage of this irregular area. For any given polygon, it can always be partitioned into convex polygon(s). In this paper, we only consider the case where these divided convex polygons are all connected one by one. In this case, the shortest path with minimum turns in each convex polygon can be easily searched by CbSPSA. And further by simply linking all of these pieces, the total coverage path for any given polygon area can be constructed. On the other hand, for a given polygon, usually there are several different cases of partition. Among the searched coverage paths for each of these different partitions, the final optimal coverage path is determined through a predefined criteria. Numerical simulation and analyses are carried out to demonstrate the effectiveness of the proposed method.

INDEX TERMS Coverage path planning (CPP), marine surveys, lawn mowing, milling, convex polygon.

I. INTRODUCTION

Coverage path planning (CPP) is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles [2], [3], [4], [5]. This CPP problem arises in various practical applications, such as vacuum cleaning robot [6], [7], [8], lawn mower [1], [9], [10], demining robot [11], [12], automated harvester [13], [14], and NC pocket machine [9], [15], [16]. These applications further can be classified into two types [1]: lawn mowing problem [9], [10], [11], [12], [13], [14] and milling problem [6], [7], [15], [16]. In the lawn mowing case, the cutter is allowed to exit the coverage area (''mow'' over non-grass area), while in the latter case, the cutter is restricted to move inside the area.

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Wang.

In this paper, we consider a marine robotics case where one or multiple marine vehicles such as UUVs and/or USVs are used to search for a sort of specific object(s) on the seafloor, e.g., a sunken ship. From the perspective of offshore operations, this is the task of coverage the search area with the shortest path and also with guaranteed sonar data quality. In the case of marine survey using sonar device, the data quality usually depends on the device's specifications. In the case of side-scan sonar, in order to compensate the nadir gap in the center area of sonar image, the vehicle's inter-track distance is usually designed as half size of the sonar swath, while it can be taken as or near to the swath width in the case of multibeam echosounder. Here the details of how to determine the inter-track distance [17], [18], [19], [20] is out of the scope of this paper, and we only consider the case where the inter-track distance is predefined and given as a constant value. On the other hand, in practice the search area is usually set by a series of way points on the map which further compose polygon(s). With the constant inter-track distance, the optimal coverage for this polygon search area usually means to design a path with minimum number of turns of the vehicles [20]. Consequently, the CPP problem considered in this paper can be formulated as follows: Given polygonal 2D search area, design a shortest way-points path with the minimum number of vehicle turns while restricting the vehicle's movement inside the polygon area. So this is a sort of milling problem.

Among the various related works published so far ([1], [9], [21], [22], [23], [24], [25] and references therein), contour-parallel milling and axis-parallel (also known as zig-zag) milling [21], [22], [23], and grid-based methods [1], [24], [25] can be considered as the most suitable candidates to solve the coverage problem for any given polygon area with no obstacle in it. In the proposed methods in these works, the cutter is commonly modelled as a circle or an axis-aligned square. Due to this kind of cutter model, the algorithms in these works might cause certain unmown or uncut area to be remained in the case of irregular polygons [9], [21]. In [9], though the authors proposed a compensation method for covering the complete area to be milled, it cannot guarantee the shortest path and moreover, in most of cases it might cause certain overlapped or crossed path pieces in the final closed path [9], [22], [23], [24], [25]. From the optimal or shortest path perspective, these overlapped or crossed paths are far from the shortest one and therefore contradict to the CPP objective in this paper.

For any given polygon, it can always be partitioned into one (the given polygon itself is convex) or several convex polygons [26], [27], [28]. For each of these convex polygons, given input and output vertices, it is easy to search a sort of shortest path with minimum number of turns using CbSPSA. Here the CbSPSA method can be characterized as: 1) Searching a sort of shortest path using calculation method, 2) full coverage of convex polygon with minimum number of turns, 3) no overlapped or crossed path pieces, and 4) all paths located inside the polygon. All of these searched paths in each of convex polygons can be easily linked to each other to compose the total coverage path for the given polygon. Here it is worth to mention that in this paper we only consider the case where the polygon can be decomposed into a series of convex polygons, each of which are connected one by one. On the other hand, according to its measurement mechanism, in this paper the sonar is modelled as a line that is perpendicular to the vehicle's forward motion and its length is the same as the inter-track distance. This kind of line model is different from the ones in the previous works where the cutter is modelled as a circle or a square. Numerical simulation and analyses are also carried out to verify the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows. Section II presents the problem statement including the polygon partition and problem formulation, and the various search algorithms in the convex polygon are described in Section III.

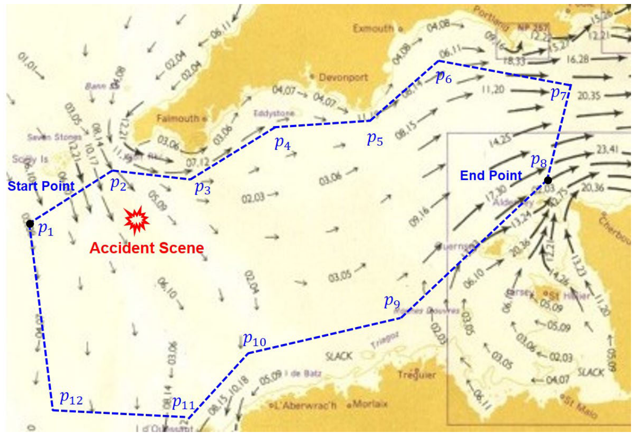


FIGURE 1. An example of search area determination in marine survey.

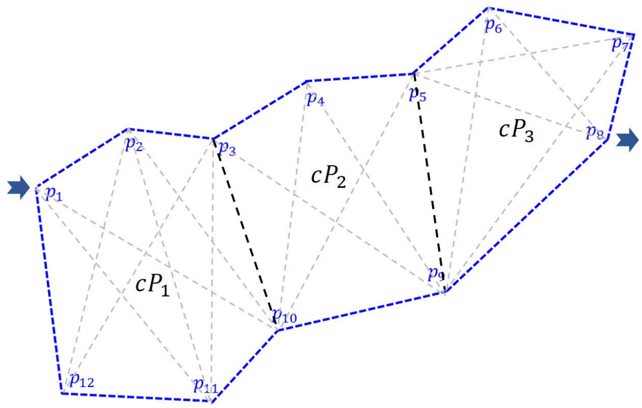


FIGURE 2. An example of polygon partition into convex polygons.

In Section IV, the coverage algorithm for partitioned convex polygon is presented while the final total coverage path planning method is proposed in Section V. Simulation studies are carried out in Section VI, and finally, a brief summary and some of future works are discussed in Section VII.

Notations: Throughout this paper, $\mathcal{P} \in \Re^{2\times n}$ indicates a general polygon with $p_i \in \Re^2$, $i = 1, \cdots, n$ its vertices and all of them are arranged clockwise, and $c\mathcal{P}$ presents a convex polygon. $V(\mathcal{P}) \in \Re^n$ denotes the vertex set of $\mathcal{P}$, and $CP(\mathcal{P}) \in \Re^{2\times N}$ means the coverage path of $\mathcal{P}$ with N the number of path way points. $||\cdot||$ indicates the Euclidean norm, $\angle ab$ denotes the azimuth angle of $\vec{ab}$. L_s denotes the inter-track distance which is the same as sonar swath width.

II. PROBLEM STATEMENT

A. POLYGON PARTITION AND PRELIMINARIES

In most of the marine surveys, the search area is usually determined by a series of way points marked on the map. By connecting these points with straight lines, then a polygonal shape of search area can be composed as seen in Fig. 1.

It is well known that for any given polygon P, it always can be partitioned into a series of convex polygons cP_i, $i = 1, \cdots$ [26], [27], [28]. As for how to partition a polygon into specific convex polygons, it is a much more complicated process [26], [27] and out of the scope of this paper.

For the convenience of discussion, in this paper we consider the relatively simple case where the polygon P has the following properties.

P1. $\mathcal{P}$ can be partitioned into a series of $c\mathcal{P}_i$ with $i = 1, \cdots, m$, all of which are connected one by one, as seen in Fig.2.

P2. $V(\mathcal{P}) = V(c\mathcal{P}_1) \cup \cdots \cup V(c\mathcal{P}_m)$.

For each $c\mathcal{P}_i$, **P1** indicates that: 1) if $1 < i < m$, then $c\mathcal{P}_i$ is connected with $c\mathcal{P}_{i-1}$ and $c\mathcal{P}_{i+1}$; 2) else, $c\mathcal{P}_1$ is only connected to $c\mathcal{P}_2$ and $c\mathcal{P}_m$ to $c\mathcal{P}_{m-1}$. **P2** presents that polygon partition in this paper does not add or eliminate any of vertex comparing to $V(\mathcal{P})$.

Suppose that the start point of $CP(\mathcal{P})$ is located at one of the vertices of $c\mathcal{P}_1$ and the end point is at the vertex of $c\mathcal{P}_m$. And moreover, the end point of $CP(c\mathcal{P}_i)$ is set as the start point of $CP(c\mathcal{P}_{i+1})$ with $i = 1, \cdots, m-1$. In this case, we have

$$CP(\mathcal{P}) = CP(c\mathcal{P}_1) + \cdots + CP(c\mathcal{P}_m). \tag{1}$$

In the remainder of this paper, $in\mathcal{P} = [i, dir]$ and $out\mathcal{P} = [j, dir]$ with $i, j = 1, \cdots, n$ and $dir = \pm 1$ indicate each of the start and end points of $CP(\mathcal{P})$, where the exact position of $x = [k, dir]$ with $k = 1, \cdots, n$ is defined as following

$$x = p_k + 0.5L_s \begin{bmatrix} cos\alpha_k \\ sin\alpha_k \end{bmatrix}, \tag{2}$$

where $\alpha_k = \angle p_i p_{i+dir}$, and $dir = 1$ indicates the clockwise direction and $dir = -1$ is vice versa.

B. PROBLEM FORMULATION

The CPP objective in this paper can be formulated as follows. For any given polygon $\mathcal{P}$ with the properties of **P1** and **P2**, and $in\mathcal{P}$ the start point and $out\mathcal{P}$ the end point of the $CP(\mathcal{P})$,

- At first, partition $\mathcal{P}$ into $c\mathcal{P}_i$ with $i = 1, \cdots, m$.
- In each $c\mathcal{P}_i$, where $inc\mathcal{P}_i = outc\mathcal{P}_{i-1}$ and $outc\mathcal{P}_i = inc\mathcal{P}_{i+1}$ with $inc\mathcal{P}_1 = in\mathcal{P}$ and $outc\mathcal{P}_m = out\mathcal{P}$, search a sort of shortest path $CP(c\mathcal{P}_i)$.
- $CP(\mathcal{P})$ is constructed by simply combining $CP(c\mathcal{P}_i)$ with $i = 1, \cdots, m$ as in (1).

III. CONVEX POLYGON AND ITS SEARCH ALGORITHMS

For any given $c\mathcal{P} \in \Re^{2\times n}$, according to its definition, it is always monotone with respect to each of its edge [29]. For any given edge $p_i p_{mod(i+1,n)}$ with $i = 1, \cdots, n$, let h_{iM} denotes the maximum vertical height corresponding to this edge. Then, the function[1] $tp = SearchSH(c\mathcal{P})$ returns a set of numbers $tp = [a, b, c]$ with $a, b, c \in \{1, \cdots, n\}$, where the vertical height from p_a to $p_b p_c$ is $h_M = min\{h_{1M}, \cdots, h_{nM}\}$. According to tp, $c\mathcal{P}$ can be rearranged as $c\mathcal{P} = p_1 \cdots p_n$ such that $p_1 = p_b$ and $p_n = p_c$. In this case, $SearchSH(c\mathcal{P})$ returns $tp = [i_M, 1, n]$, $i_M < n$. For a vertex p_i, $i \in \{1, \cdots, n\}$, if $1 \leq i \leq i_M$, then call it is in *LEFT*, else if $i_M \leq i \leq n$, then it's in *RIGHT*. The main idea of minimizing the vehicle's turn numbers in this paper is that: Force the vehicle to search the area along the tracks which are paralleled to the edge $p_1 p_n$ so as for the track number $ceil(h_M/L_s)$ to be minimized.

[1] All the functions referred in this paper are developed by the authors.

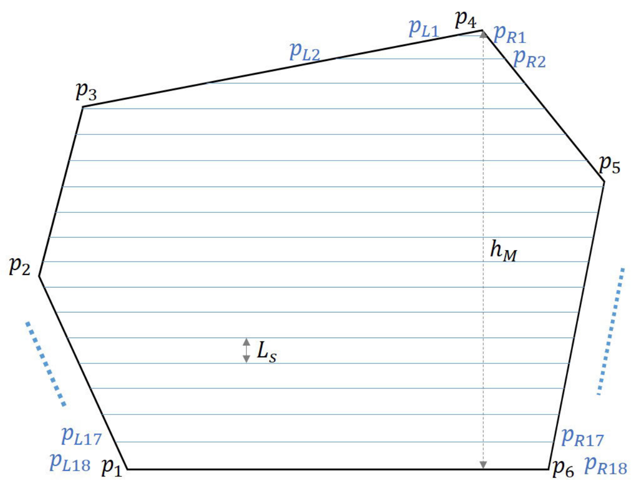


FIGURE 3. Convex polygon partition with tracks paralleled with $p_1 p_n$.

Considering $c\mathcal{P}$ as in Fig. 3 with $n_c = floor(h_M/L_s)$, $c\mathcal{P}$ can be partitioned by a series of trapezoids and one remainder polygon $p_{L1}p_4p_{R1}$ as seen in Fig. 3. In each trapezoid $p_{Li}p_{Ri}p_{Ri+1}p_{Li+1}$, it is common to design a coverage path as the center line which is parallel to the line $p_{Li}p_{Ri}$. However, by simply connecting these of neighboring center lines [1], [9], [21], [22], [23], [24], [25], usually we cannot guarantee the full coverage of the area, especially in the corner area near to $p_{Li}p_{Li+1}$ or $p_{Ri}p_{Ri+1}$. In order to solve the coverage problem in these (irregular) corner areas with the guarantee of a sort of shortest path, there are various functions (algorithms) coded and applied in this paper, among which some of the most commonly used ones are described as follows.

A. $[x_M, b_T] = calPoint1(X_1, X_2, X_0)$

For any given three points X_0, X_1, X_2, as seen in Fig. 4, the function $calPoint1(X_1, X_2, X_0)$ is to find a shortest path from X_1 to X_2 which also cover the point X_0 by the sonar. Let h denotes the vertical distance from X_0 to line X_1X_2. If $h \leq R$, then set $b_T = 0$, otherwise set $b_T = 1$. In the latter case, the function further returns the point x_M (it's 2D coordinate) such that $||X_1x_M|| + ||x_MX_2|| \leq ||X_1x|| + ||xX_2||$ for all $x = X_0 + R[cos\alpha; sin\alpha]$ with $\alpha_1 \leq \alpha \leq \alpha_2$.

The point x_M can be determined mathematically, also can be acquired through calculation method. If we set $\alpha = \alpha_1 + k\Delta\alpha$ where $k = 1 : K$ with $K = floor(abs(\alpha_2 - \alpha_1)/\Delta\alpha)$, then x_M is the point corresponding to x_{α_k} such that $||X_1x_{\alpha_k}|| + ||x_{\alpha_k}X_2||$ is the minimum value. Here $\Delta\alpha$ is a design parameter. The smaller it is the more accurate the calculation result. However, too small value of $\Delta\alpha$ might significantly increase the calculation load.

B. POINTS=$calPoints4(X_1, X_2, \beta)$

Let's consider Fig. 5, where X_1 is located between l_0 and l_1 and X_2 on the l_2, also $\beta = \angle l_1$. In this case, $calPoints4(X_1, X_2, \beta)$ returns the points x_{o1} and x_{o2}, both of

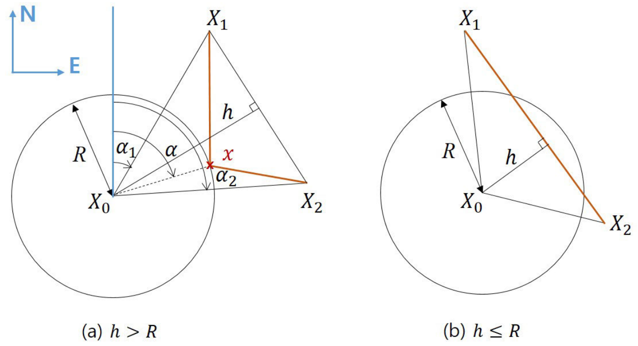


FIGURE 4. Illustration of *calPoint*1(X_1, X_2, X_0), (a) with the case $b_T = 1$, (b) the case of $b_T = 0$.

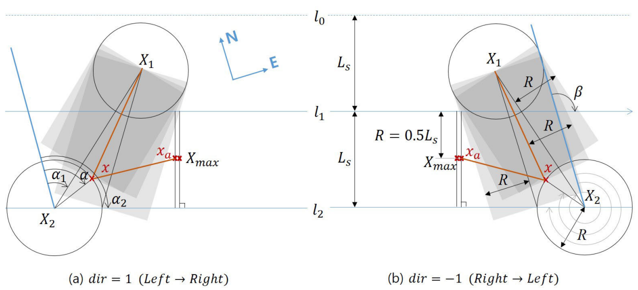


FIGURE 5. Illustration of *calPoints*4(X_1, X_2, β), (a) with $dir = 1$ (Left→Right), (b) $dir = -1$ (Right→Left).

which satisfy that $||X_1x_{o1}|| + ||x_{o1}x_{o2}|| + ||x_{o2}X_{max}|| \le ||X_1x|| + ||xx_a|| + ||x_aX_{max}||$ for all $x = X_2 + R[cos\alpha; sin\alpha]$. Here, if $dir = 1$, then $\alpha_1 \le \alpha \le \alpha_2$; else if $dir = -11$, then $\alpha_1 \ge \alpha \ge \alpha_2$. This function also can be solved by using calculation method. If we set $\alpha = \alpha_1 + dir \cdot k\Delta\alpha$ with $k = 1 : K$ and $K = floor(abs(\alpha_2 - \alpha_1)/\Delta\alpha)$, then it is easy to search the minimum value of $||X_1x_{\alpha_k}|| + ||x_{\alpha_k}x_{a,\alpha_k}|| + ||x_{a,\alpha_k}X_{max}||$ and further has $x_{o1} = x_{\alpha_k}$ and $x_{o2} = x_{a,\alpha_k}$.

If there is a vertex p_k between l_0 and l_2 and its vertical distance to X_1x is larger than R, then the point $x' = calPoint1(X1, x, p_k)$ will be added to the way points such that the returned path becomes $x'x_{o1}x_{o2}$.

C. POINTS=calPoints5(X_1, X_2, α)

As seen in Fig. 6, if $\alpha \ge \pi/2$, then this function returns x_1; otherwise returns x_1 and x_2. Here $x_2 = X_1 + R[cos\angle X_1x_1; sin\angle X_1x_1]$.

D. POINTS=calPoints7(X_1, X_2, β, DIR)

Let's consider Fig. 7. If $\alpha \le \pi/4$, then it returns x_1x_2 (if $dir = 1$) or x'_1x_2 (if $dir = -1$); otherwise returns $x_1x_3x_2$ (if $dir = 1$) or $x'_1x'_3x_2$ (if $dir = -1$). Here $x_3 = calPoint1(x_1, x_2, X_2)$ and $x'_3 = calPoint1(x'_1, x_2, X_2)$.

E. POINTS=calPoints9(X, α, β, DIR)

For given X, α, and β as seen in Fig. 8, it is easy to calculate x_1, x_2, and x_3. Here, $x = X + L_s[cos\gamma; sin\gamma]$ with $\gamma = \pi + \beta + \alpha/2$, $x_1 = x + R[cos\gamma_1; sin\gamma_1]$ with $\gamma_1 = \beta + \pi/2$,

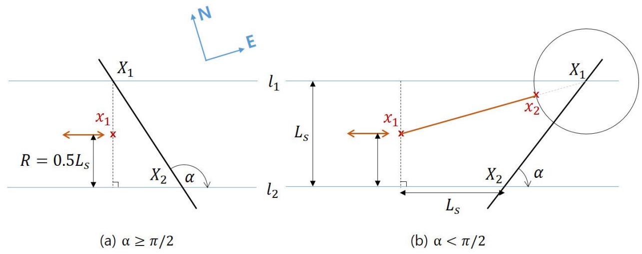


FIGURE 6. Illustration of *calPoints*5(X_1, X_2, α), (a) with $\alpha \ge \pi/2$, (b) $\alpha < \pi/2$.

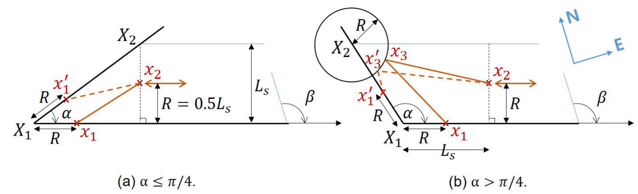


FIGURE 7. Illustration of *calPoints*7(X_1, X_2, β, dir), (a) with $\alpha \le \pi/4$, (b) $\alpha > \pi/4$.

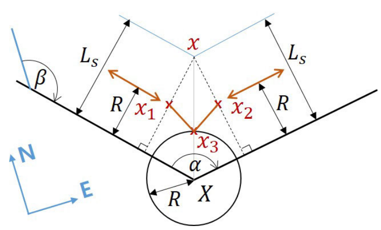


FIGURE 8. Illustration of *calPoints*9(X, α, β, dir).

and $x_2 = x + R[cos\gamma_2; sin\gamma_2]$ with $\gamma_2 = \alpha + \beta - \pi$. Further, we have $x_3 = calPoint1(x_1, x_2, X)$. If $dir = 1$, then this function returns $x_1x_3x_2$, else if $dir = -1$ returns $x_2x_3x_1$.

IV. CbSPSA FOR $CP(c\mathcal{P}_i)$

A. FURTHER PARTITION OF $c\mathcal{P}_i$

For given $c\mathcal{P}_i$, suppose $inc\mathcal{P}_i = [j, dir]$, $outc\mathcal{P}_i = [k, dir]$ and $SearchSH(c\mathcal{P}_i) = [i_M, 1, n]$ with $i_M < n$. In the case of $j, k \in \{1, i_M, n\}$, $c\mathcal{P}_i$ can be directly applied to search for $CP(c\mathcal{P}_i)$. However, in the case where $j, k \notin \{1, i_M, n\}$ or $j \notin 1, i_M, n \cup k \notin 1, i_M, n$, we have to additional process that further partitioning this convex polygon into maximum of three parts: $inLet$, $c\mathcal{P}_i^c$, and $outLet$. If $j \in \{1, i_M, n\}$, then no need to partition $inLet$; or $k \in \{1, i_M, n\}$, then $outLet$ can be omitted.

Consider an example as shown in Fig. 9, where $c\mathcal{P}_i = p_1p_2p_3p_4p_5p_6$ with $SearchSH(c\mathcal{P}_i) = [4, 1, 6]$ and $inc\mathcal{P}_i = [3, -1]$, $outc\mathcal{P}_i = [5, 1]$. At first, we calculate and compare two components $A = ||p_3p_4|| + ||p_5p_6||$ and $B = ||p_3p_2|| + ||p_2p_1|| + ||p_5p_4||$. If $B < A$ as in this example, we set $inc\mathcal{P}_i$ to be linked to p_1 and $outc\mathcal{P}_i$ to p_4, as seen in Fig. 9; else if $A \le B$, then p_3 will be linked to p_4 to construct $inLet$, and $outLet$ will link $p_5 \to p_6$. Consequently, according to Fig. 9,

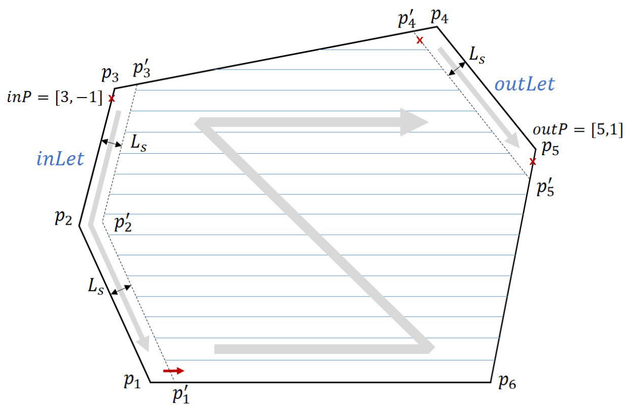


FIGURE 9. An example of convex polygon partition.

$CP(c\mathcal{P}_i)$ can be constructed as

$$CP(c\mathcal{P}_i) = CP(inLet) + CP(c\mathcal{P}_i^c) + CP(outLet), \quad (3)$$

where $c\mathcal{P}_i^c = p_1'p_2'p_3'p_4'p_5'p_6$. Here *inLet* and *outLet* are constructed by a shrinking process. *inLet* is made by moving the edges p_1p_2 and p_2p_3 toward their interior in a self-parallel manner [30] with the moving distance as L_s, and *outLet* is by p_4p_5.

In order for (3) to be held, here we need the following assumption.

Assumption 1: In (3), it satisfies that $SearchSH(c\mathcal{P}_i) = SearchSH(c\mathcal{P}_i^c)$.

Suppose h denotes the vertical distance from p_4' to $p_1'p_6$ and h_M is the minimum height of $c\mathcal{P}_i$ while h_M' is of $c\mathcal{P}_i^c$. It is easy to verify that $h_M' \le h < h_M$. If *Assumption 1* is held, then we have $h = h_M'$. In this case, $CP(c\mathcal{P}_i^c)$ searched by as in Fig. 9 can guarantee the minimum number of turns. However, it's notable that even if *Assumption 1* is not satisfied, since $h_M - h_M' < 2L_s$, we have $h - h_M' < 2L_s$, which means that $CP(c\mathcal{P}_i^c)$ as in Fig. 9 has a maximum of one more turn than the possible minimum turns.

While searching for $CP(inLet)$ and $CP(outLet)$, two functions *calPoints*7 and *calPoints*9 are mainly used to search for the way points. For example, searching for near the corner of p_3p_3', p_4p_4', and p_5p_5', *calPoints*7 is used while *calPoints*9 is applied in the case of p_1p_1' and p_2p_2'. Here it is worth to mention that in *inLet*, the function *calPoints*7 returns $x_1x_2(x_1'x_2)$ or $x_1x_3x_2(x_1'x_3'x_2)$, and in *outLet* it returns $x_2x_1(x_2x_1')$ or $x_2x_3x_1(x_2x_3'x_1')$.

B. CbSPSA FOR $CP(c\mathcal{P}_i^c)$

Let us still consider the example as seen in Fig. 9. Suppose $n_c' = floor(h_M'/L_s)$, then $c\mathcal{P}_i^c$ can be divided into n_c' number of trapezoids and $c\mathcal{P}_{i,init}^c$. Here $c\mathcal{P}_{i,init}^c$ can be a triangle $p_4p_{R1}p_{L1}$ (or a convex polygon, if there are vertices between p_4 and p_{L1} or/and p_4 and p_{R1}). The search methods in each trapezoid are similar except in the last one $p_{Ln_c'}p_{Rn_c'}p_6p_1$. Consequently, CbSPSA for $CP(c\mathcal{P}_i^c)$ can be summarized as the pseudo-codes in Algorithm 1 and 2.

Algorithm 1 *Search_Polygon*$(c\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s)$

Input: $c\mathcal{P},\ inc\mathcal{P},\ outc\mathcal{P},\ L_s$
Output: $CP(c\mathcal{P})$

```
CP(cP) ← ∅
n_c = floor(h_M/L_s)
for i=1:n_c
  if i==1
    PPoints = Search_initPolygon(cP_init, stat)
    CP(cP) ←(add) PPoints
  else
    X_1 :=last way point of CP(cP)
    X_2 := p_Li(if dir = 1) or p_Ri(if dir = −1)
    PPoints = calPoint4(X_1, X_2, dir)
    CP(cP) ←(add) PPoints
    if i < n_c
      X_1 := p_R(i−1)(if dir = 1) or p_L(i−1)(if dir = −1)
      X_2 := p_Ri(if dir = 1) or p_Li(if dir = −1)
      PPoints = calPoints5(X_1, X_2, ∠p_1p_n)
    else
      X_1 := p_Rn_c(if dir = 1) or p_Ln_c(if dir = −1)
      X_2 := p_n(if dir = 1) or p_1(if dir = −1)
      PPoints = calPoints7(X_1, X_2, ∠p_1p_n, dir)
    end if
    CP(cP) ←(add) PPoints
  end if
end for
return CP(cP)
```

C. CbSPSA FOR $CP(c\mathcal{P}_i)$

According to (3), the coverage path for $c\mathcal{P}_i$ can be easily searched as Algorithm 3, where the function *pPolygon*2 partitions a given convex polygon $c\mathcal{P}$ into a maximum of three parts, and returns $c\mathcal{P}^c$ as well as the coverage path of $CP(inLet)$ and $CP(outLet)$.

V. CbSPSA FOR $CP(\mathcal{P})$

As mentioned before, for any given polygon, how to partition it into desirable convex polygons is a much more complicated process [26], [27]. In this paper we only consider relatively simple cases where the polygon satisfies **P1** and **P2**. Moreover, with some of suitable additional conditions, it is possible to sort out all of possible partitions for a given polygon. Recall the example case in Fig. 2. In this case, if we add the condition that the number of partitioned convex polygons should be as minimum as possible, then it is available for us to sort all the possible partitioned cases. With these conditions, in the case of Fig. 2, it is easy to see that there are total of 5 cases as seen in Fig. 10.

For each case, the total coverage path $CP(\mathcal{P})$ can be searched as (1) with each $CP(c\mathcal{P}_j)$, $j \in \{1, 2, 3\}$ searched by Algorithm 3. For any given polygon $\mathcal{P}$ and its total of n_p cases of possible convex polygon partitions, we define the

Algorithm 2 *Search_initPolygon*$(c\mathcal{P}_{init}, stat)$

Input: $c\mathcal{P}_{init}, stat$
Output: $CP(c\mathcal{P}_{init})$
1 $CP(c\mathcal{P}_{init}) \leftarrow \emptyset$
2 $x_1 := p_{iM} + R[cos\gamma; sin\gamma]$ with γ depends on $stat[0] = 1$
3 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_1$
4 **if** x_1 is on the edge of $c\mathcal{P}_{init}$
5 **if** path direction is *Left*$\rightarrow$ *Right*
6 $[x_2, b_T] = calPoint1(x_1, p_{R1}, p_{L1})$
7 **if** $b_T == 1$
8 $[x_3, b_T] = calPoint1(x_1, x_2, p_{iM-1})$
9 **if** $b_T == 1$
10 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_3$
11 **end if**
12 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_2$
13 $x_4 := p_{R1} + R[cos\angle p_{R1}x_2; sin\angle p_{R1}x_2]$
14 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_4$
15 **else**
16 $x2 := p_{R1} + R[cos\angle p_{R1}x_1; sin\angle p_{R1}x_1]$
17 $[x_3, b_T] = calPoint1(x_1, x_2, p_{iM+1})$
18 **if** $b_T == 1$
19 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_3$
20 **end if**
21 $CP(c\mathcal{P}_{init}) \overset{add}{\leftarrow} x_2$
22 **end if**
23 **elseif** path direction is *Right*$\rightarrow$ *Left*
24 *Similar to 6–21*
25 **end if**
26 **end if**
27 **return** $CP(c\mathcal{P}_{init})$

Algorithm 3 $CP(c\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s)$

Input: $c\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s$
Output: $CP(c\mathcal{P})$
1 $CP(c\mathcal{P}) \leftarrow \emptyset$
2 $[c\mathcal{P}^c, CP(inLet), CP(outLet)] = pPolygon2(c\mathcal{P}, inc\mathcal{P},$
3 $outc\mathcal{P}, L_s)$
4 $CP(c\mathcal{P}^c) = Search\_Polygon(c\mathcal{P}^c, inc\mathcal{P}, outc\mathcal{P}, L_s)$
5 $CP(c\mathcal{P}) = CP(inLet) + CP(c\mathcal{P}^c) + CP(outLet)$
6 **return** $CP(c\mathcal{P})$

following criteria function as

$$f^c[CP(\mathcal{P})_k] = \gamma_{tL} \cdot \frac{tL_k}{tL_M} + \gamma_{tN} \cdot \frac{tN_k}{tN_M}, \quad k = 1, \cdots, n_p \quad (4)$$

where tL_k is the total length of $CP(\mathcal{P})_k$ and tN_k the total numbers of the vehicle's turns, and the subscript in both of tL_M and tN_M indicates the corresponding maximum value among $k = 1, \cdots, n_p$. Two weighting factors γ_{tL} and γ_{tN} are normalized as $\gamma_{tL} + \gamma_{tN} = 1$.

CbSPSA for $CP(\mathcal{P})$ becomes clear that: we search each of $CP(\mathcal{P}_k)$ and, according to the calculated $f^c[CP(\mathcal{P})_k]$, choose the case with the minimum value of it, as seen in Algorithm 4. The function $pPolygon1$ partitions the given polygon $\mathcal{P}$ into total of n_p possible cases so that $\mathcal{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_{n_p}\}$. For all $\mathcal{G}_i$, they have the same number of the partitioned convex polygons.

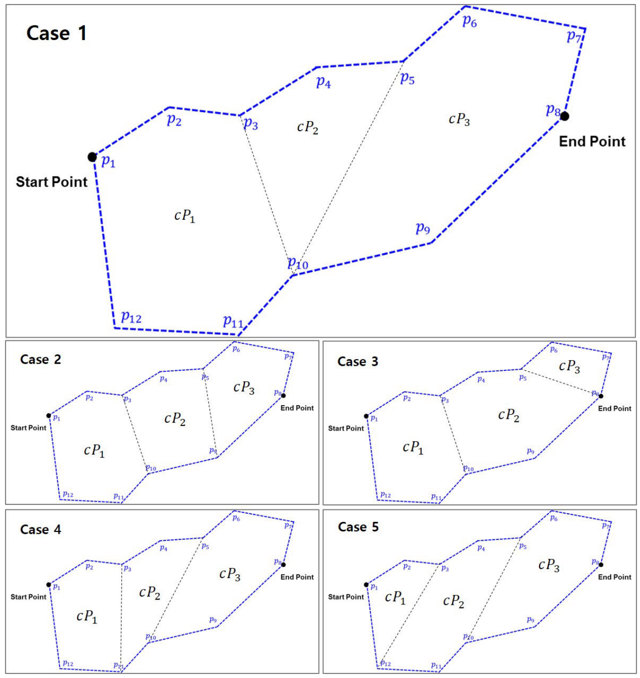


FIGURE 10. Total of 5 possible cases of polygon partition.

Algorithm 4 $CPP(\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s)$

Input: $\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s$
Output: $CP(\mathcal{P})$
1 $CP(\mathcal{P}) \leftarrow \emptyset$
2 $[n_p, \mathcal{G}] = pPolygon1(\mathcal{P}, inc\mathcal{P}, outc\mathcal{P}, L_s)$
3 **for** i=1:n_p
4 $CP(\mathcal{G}_i) \leftarrow \emptyset$
5 **for** j=1:n_i
6 $CP(\mathcal{G}_i) \overset{add}{\leftarrow} CP(c\mathcal{P}_j, inc\mathcal{P}, outc\mathcal{P}, L_s)$
7 **end for**
8 $c_i := f^c[CP(\mathcal{G}_i)]$
9 **if** $c_i < c_{i-1}$
10 $CP(\mathcal{P}) = CP(\mathcal{G}_i)$
11 **end if**
12 **end for**
13 **return** $CP(\mathcal{P})$

VI. NUMERICAL STUDIES

In this section, some of numerical studies are carried out to verify the effectiveness of the proposed coverage path planning method. The algorithm is implemented in MATLAB and we consider the example case shown in Fig. 2, where $\mathcal{P}$ = [2850,275; 3575,1400; 3450,2450; 4175,3625; 4250,4950; 5075,5800; 4975,7650; 3425,7075; 1575,5350; 1100,3350; 200,2425; 300,550]^T with the length unit as m. The inter-track distance is set as $L_s = 160m$.

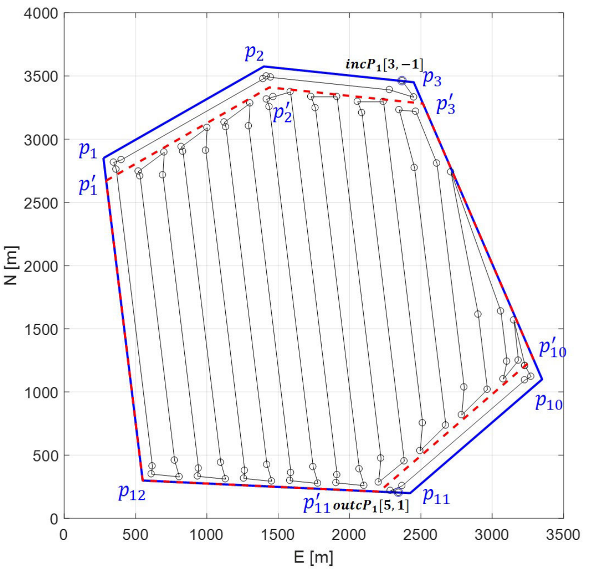


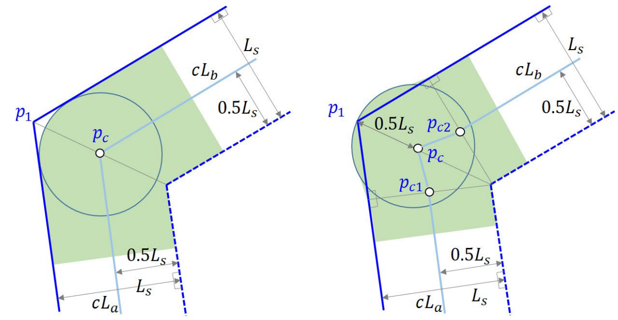
FIGURE 11. Convex polygon partition and searched coverage path $CP(c\mathcal{P}_1)$.

A. CbSPSA FOR $CP(c\mathcal{P}_1)$

At first, we consider a single convex polygon $c\mathcal{P}_1 = p_1p_2p_3p_{10}p_{11}p_{12}$,where the vertexes are as defined in Fig. 2. In the case with $inc\mathcal{P}_1 = [3, -1]$ and $outc\mathcal{P}_1 = [5, 1]$, the searched coverage path for this convex polygon using Algorithm 3 is shown in Fig. 11, where black circles present the coverage path way points.

For given convex polygon $c\mathcal{P}_1 = p_1p_2p_3p_{10}p_{11}p_{12}$, search function $tp = SearchSH(c\mathcal{P}_1)$ returns $tp = [10, 1, 12]$. This means that the vertical height h_{10M} from p_{10} to the edge p_1p_{12} is the shortest one among h_{iM}, $i = 1, 2, 3, 10, 11, 12$. Therefore, if we design the tracks paralleling to the edge p_1p_{12}, then we can get the minimum number of tracks $n_c = floor(h_{10M}/L_s)$ as described in Section III. Then, as seen in Algorithm 3, we apply the function $pPolygon2(c\mathcal{P}_1, inc\mathcal{P}_1, outc\mathcal{P}_1, L_s)$ and partition the polygon into three parts: $c\mathcal{P}_1^c$, $inLet$, and $outLet$. This function further return the information of convex polygon $c\mathcal{P}_1^c$, and the coverage paths, $CP(inLet)$ and $CP(outLet)$. And for remained $c\mathcal{P}_1^c$, we apply the algorithm $Search_Polygon(c\mathcal{P}_1^c, inc\mathcal{P}_1^c, outc\mathcal{P}_1^c, L_s)$ to search its coverage path. Here $inc\mathcal{P}_1^c$ is the end point of $CP(inLet)$ and $outc\mathcal{P}_1^c$ is the start point of $CP(outLet)$, respectively. Finally, the coverage path for $c\mathcal{P}_1$ can be constructed as $CP(c\mathcal{P}_1) = CP(inLet) + CP(c\mathcal{P}_1^c) + CP(outLet)$ as seen in Algorithm 3.

In order to investigate the full coverage of given polygon, especially in the corner area, we scale up the area near the vertex p_1 as seen in Fig. 12. If we directly connect the two center lines to construct the coverage path, which is the generally used method in the related works published so far, then it's not difficult to find that there is an area near vertex remained uncovered. To solve this problem, in this paper, we construct the path $p_{c1}p_cp_{c2}$ (see Fig.12) so that can guarantee the full coverage at the corner area.



(a) Directly connecting the center lines. (b) Proposed method.

FIGURE 12. Full coverage at the corner area of irregular polygon.

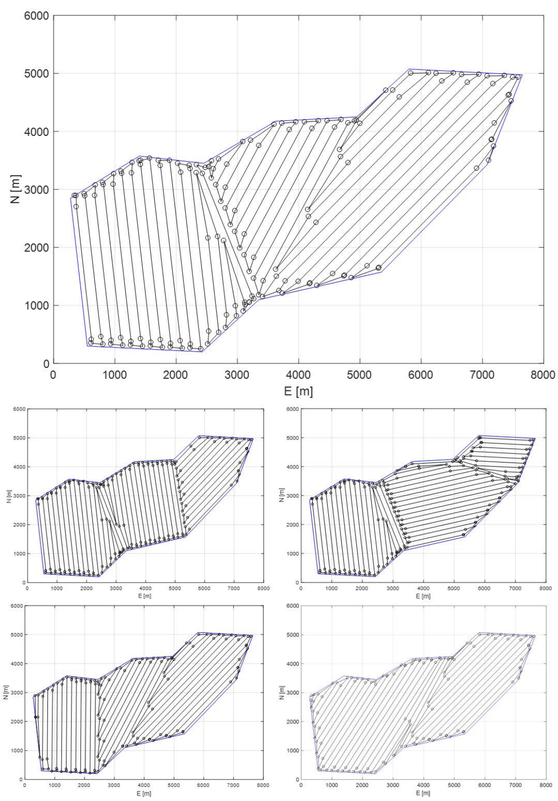


FIGURE 13. $CP(\mathcal{G}_k)$ with $k = 1, \cdots, 5$.

B. CbSPSA FOR $CP(\mathcal{P})$

For given $\mathcal{P}$ (see Fig. 10), suppose $inc\mathcal{P} = [1, -1]$ and $outc\mathcal{P} = [8, 1]$. As mentioned in Section IV, if we set the condition that the number of partitioned convex polygons should be as minimum as possible, then it is easy to sort out all the possible partition cases, which is as seen in Fig. 10. For these total of 5 cases of convex polygon partitions, searched coverage paths for each of them are as shown in Fig. 13, and calculated tL_k, tN_k, and corresponding $f^c[CP(\mathcal{G}_k]$ are shown in Table 1. In the simulation, we set $\gamma_{tL} = 0.4$ and $\gamma_{tN} = 0.6$. From Table 1, we can see that *Case 5*, though it has the longest path length, but also has the minimum of $f^c[CP(\mathcal{G}_5)]$ due to it's minimum number of turns. And Fig. 14

TABLE 1. Comparison of $CP(\mathcal{G}_k)$ with $k = 1, \cdots, 5$.

Parameters	$case1$	$case2$	$case3$	$case4$	$case5$
tL_k $[km]$	137.93	134.254	136.708	139.651	140.262
tN_k	44	48	49	40	37
$f^c[CP(\mathcal{G}_k)]$	0.9321	0.9706	0.9899	0.8881	0.8531

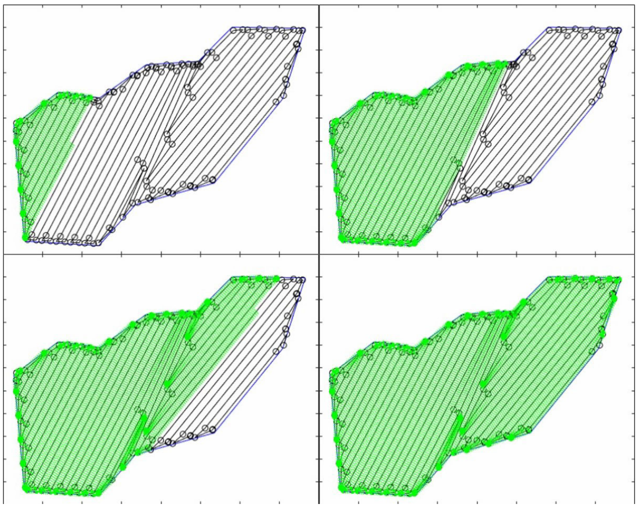


FIGURE 14. Coverage of irregular polygon area using the sonar mounted on the vehicle.

shows the searched coverage path of *Case 5* and its coverage by the sonar mounted on the marine vehicle. In the simulation, we set the sonar swath width as L_s and it is easy to see that the sonar can coverage the full area of $\mathcal{P}$ while the vehicle is moving along the $CP(\mathcal{P})$.

It is worth to mention that the design of the parameters tL_k and tN_k is dependent on the specific requirement in the practical applications. If time is critical, then we have to increase γ_{tL} while decreasing γ_{tN}; and if the mission is to acquire detailed and high quality seabed images, then it's better to increase γ_{tN}.

VII. CONCLUSION

This paper has present a novel coverage path planning algorithm for full coverage of irregular polygon area. For any given polygon, it can always be partitioned into a series of convex polygons. In each of these convex polygons, the proposed coverage algorithm CbSPSA can easily search a sort of shortest coverage path, and further under the assumption that all these convex polygons are connected one by one, the overall coverage path can be constructed by simply linking all these paths searched in each convex area one by one. Numerical studies also have been carried out to illustrate the effectiveness of the proposed method.

Here it's worth to mention some of our interesting future works. First, in this paper we only consider relatively simple cases where given polygon always can be partitioned into a series of convex polygons which are further connected one by one. In fact, polygon decomposition (or partition) is quite a complicated problem. Therefore, how to decompose a given polygon and further to integrate all the coverage paths searched in each convex polygon under more relaxed conditions might be one of most interesting research issue.

On the other hand, this paper does not consider the marine vehicle's detailed practical and operational issues. For most of underwater survey vehicles, they have torpedo-like mechanical structure and therefore have their own minimum turn radius. Another main practical issue is that how to properly deal with the sea current during the survey. All of these issues also should be considered in our upcoming sea trials, which are scheduled in the early next year.

REFERENCES

[1] E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell, "Approximation algorithms for lawn mowing and milling," *Comput. Geometry*, vol. 17, nos. 1–2, pp. 25–50, Oct. 2000.
[2] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, pp. 113–126, Oct. 2001.
[3] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.
[4] R. Bormann, F. Jordan, J. Hampp, and M. Hägele, "Indoor coverage path planning: Survey, implementation, analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QL, Australia, May 2018, pp. 1718–1725.
[5] T. Cabreira, L. Brisolara, and P. R. Ferreira, "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, Jan. 2019, doi: 10.3390/drones3010004.
[6] F. Yasutomi, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1988, pp. 1839–1841.
[7] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *Robot. Auto. Syst.*, vol. 14, nos. 2–3, pp. 199–212, May 1995.
[8] X. Miao, J. Lee, and B.-Y. Kang, "Scalable coverage path planning for cleaning robots using rectangular map decomposition on large environments," *IEEE Access*, vol. 6, pp. 38200–38215, 2018.
[9] Y. Y. Huang, Z. L. Cao, S. J. Oh, E. U. Kattan, and E. L. Hall, "Automatic operation for a robot lawn mower," *Proc. SPIE*, vol. 727, pp. 344–354, Feb. 1987.
[10] M. Höffmann, J. Clemens, D. Stronzek-Pfeifer, R. Simonelli, A. Serov, S. Schettino, M. Runge, K. Schill, and C. Büskens, "Coverage path planning and precise localization for autonomous lawn mowers," in *Proc. 6th IEEE Int. Conf. Robotic Comput. (IRC)*, Dec. 2022, pp. 238–242.
[11] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *Int. J. Robot. Res.*, vol. 22, nos. 7–8, pp. 441–466, 2003.
[12] M. Dakulovic and I. Petrovic, "Complete coverage path planning of mobile robots for humanitarian demining," *Ind. Robot*, vol. 39, no. 5, pp. 484–493, Aug. 2012.
[13] A. Stoll and H. D. Kutzbach, "Guidance of a forage harvester with GPS," *Precis. Agricult.*, vol. 2, pp. 281–291, Nov. 2000.
[14] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. Field Robot.*, vol. 26, no. 8, pp. 651–668, Aug. 2009.
[15] G. Vosniakos and P. Papapanagiotou, "Multiple tool path planning for NC machining of convex pockets without islands," *Robot. Comput.-Integr. Manuf.*, vol. 16, no. 6, pp. 425–435, Dec. 2000.
[16] X. Chen, T. M. Tucker, T. R. Kurfess, R. Vuduc, and L. Hu, "Max orientation coverage: Efficient path planning to avoid collisions in the CNC milling of 3D objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 6862–6869.
[17] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1827–1838, Dec. 2013.
[18] B. Sun, D. Zhu, C. Tian, and C. Luo, "Complete coverage autonomous underwater vehicles path planning based on Glasius bio-inspired neural network algorithm for discrete and centralized programming," *IEEE Trans. Cognit. Develop. Syst.*, vol. 11, no. 1, pp. 73–84, Mar. 2019.

[19] V. Yordanova and B. Gips, "Coverage path planning with track spacing adaptation for autonomous underwater vehicles," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4774–4780, Jul. 2020.
[20] A. Bagnitckii, A. Inzartsev, and A. Pavin, "Planning and correction of the AUV coverage path in real time," in *Proc. IEEE Underwater Technol. (UT)*, Feb. 2017, pp. 1–6, doi: 10.1109/UT.2017.7890299.
[21] M. Held. *On the Computational Geometry of Pocket Maching* (Lecture Notes in Computer Science), vol. 500. New York, NY, USA: Springer, 1991.
[22] S. Dhanik and P. Xirouchakis, "Contour parallel milling tool path generation for arbitrary pocket shape using a fast marching method," *Int. J. Adv. Manuf. Technol.*, vol. 50, nos. 9–12, pp. 1101–1111, Oct. 2010.
[23] R. Venkatesh, V. Vijayan, A. Parthiban, T. Sathish, and S. S. Chandran, "Comparison of different tool path pocket milling," *Int. J. Mech. Eng. Technol.*, vol. 9, no. 12, pp. 922–927, 2018.
[24] S. C. Wong and B. A. MacDonald, "A topological coverage algorithm for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2003, pp. 1685–1690.
[25] J. Tang, C. Sun, and X. Zhang, "MSTC*: Multi-robot coverage path planning under physical constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, Xi'an, China, May/Jun. 2021, pp. 2518–2524.
[26] S. Hertel and K. Mehlhorn, "Fast triangulation of the plane with respect to simple polygons," *Inf. Control*, vol. 64, Issues nos. 1–3, pp. 52–76, 1985.
[27] R. Nandakumar and N. Ramana Rao, "Fair partitions of polygons: An elementary introduction," *Proc. Indian Acad. Sci.*, vol. 122, no. 3, pp. 459–467, Aug. 2012.
[28] S. Brown and S. L. Waslander, "The constriction decomposition method for coverage path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 3233–3238.
[29] P. S. Heckbert, *Testing the Convexity of a Polygon*. London, U.K.: Academic Press, 1994.
[30] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner, "A novel type of skeleton for polygons," *J. Universal Comput. Sci.*, vol. 1, no. 2, pp. 752–761, 1996.

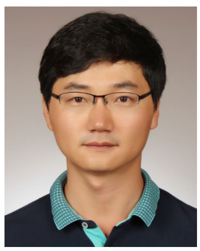
JI-HONG LI (Senior Member, IEEE) received the B.S. degree in physics from Jilin University, China, in 1991, and the M.E. and Ph.D. degrees in electronics engineering from Chungnam National University, Daejeon, South Korea, in 1999 and 2003, respectively. Currently, he is a Chief Researcher with the Korea Institute of Robotics and Technology Convergence, Pohang, South Korea; an Adjunct Professor with Pukyong National University, Busan, South Korea; and a Guest Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences, China. He has published more than 150 peer-reviewed papers and received several best paper awards in the academic conferences. Also, he is the Board Member of the Korea Marine Robot Technology Society and Korea Institute of Unmanned Systems, and a member of IFAC TC2.3 and TC7.2. His current research interests include the navigation, guidance, and control of various underwater vehicles.

HYUNGJOO KANG received the B.S. and M.S. degrees in robotics engineering from Tongmyong University, Busan, South Korea, in 2012 and 2014, respectively. He is currently a Senior Researcher with the Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, South Korea. His current research interests include marine robotics, system integration (SI), identification, modeling, and the control of various marine vehicles.

MIN-GYU KIM received the B.S. degree in mechanical engineering from the Kumoh National Institute of Technology (KIT), Gumi, South Korea, in 2014, and the M.S. degree in mechanical engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2022. Currently, he is a Senior Researcher with the Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, South Korea. His current research interests include the design and analysis of various marine robotics.

HANSOL JIN received the B.S. and M.S. degrees in mechanical engineering from Korea Maritime and Ocean University, Busan, South Korea, in 2019 and 2021, respectively. He is currently a Researcher with the Korea Institute of Robotics and Technology Convergence, Pohang, South Korea. His current research interests include marine robotics, camera, and sonar image processing based underwater simultaneous localization and mapping (SLAM).

MUN-JIK LEE received the B.S. degree in mechatronics engineering from the Tech University of Korea, Siheung, South Korea, in 2005, and the M.S. degree in control and instrumentation engineering from Kyungpook National University, Daegu, South Korea, in 2013. Currently, he is a Senior Researcher with the Korea Institute of Robotics and Technology Convergence, Pohang, South Korea. His current research interests include underwater power, communication, and electrical and electronic systems, hydraulic systems, and control.



GUN RAE CHO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2001, 2003, and 2010, respectively. He was a Senior Engineer with Samsung Heavy Industries, from 2010 to 2016. He is currently a Chief Researcher with the Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, South Korea. His current research interests include marine robotics, robust control, manipulation, and reinforcement learning.

CHULHEE BAE received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Kongju National University, Kongju, South Korea, in 2016, 2018, and 2023, respectively. He is currently a Senior Researcher with the Korea Institute of Robotics and Technology Convergence (KIRO), Pohang, South Korea. His current research interests include distance-based SLAM, 3D point cloud recognition, and acoustic source localization for marine robots.