J Electr Eng Technol Vol. 7, No. 6: 1023-1033, 2012
http://dx.doi.org/10.5370/JEET.2012.7.6.1023

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Optimal Underwater Coverage of a Cellular Region by Autonomous Underwater Vehicle Using Line Sweep Motion

Myoung Hwan Choi†

Abstract – An underwater planar covering problem is studied where the coverage region consists of polygonal cells, and line sweep motion is used for coverage. In many subsea applications, sidescan sonar has become a common tool, and the sidescan sonar data is meaningful only when the sonar is moving in a straight line. This work studies the optimal line sweep coverage where the sweep paths of the cells consist of straight lines and no turn is allowed inside the cell. An optimal line sweep coverage solution is presented when the line sweep path is parallel to an edge of the cell boundary. The total time to complete the coverage task is minimized. A unique contribution of this work is that the optimal sequence of cell visits is computed in addition to the optimal line sweep paths and the optimal cell decomposition.

Keywords: Autonomous underwater vehicle, Underwater coverage, Line sweep motion, Optimal coverage, Sidescan sonar, Group traveling salesman problem

1. Introduction

The coverage problem in the field of robotics is the problem of moving a sensor over a given region. It has been utilized in floor cleaning [1, 2], lawn mowing [3], mine hunting [4], automated harvesters [5], and window cleaners [6]. Subsea application of the coverage problem includes autonomous underwater coverage [7], mine detection and classification in mine countermeasure tasks [8, 9]. Sidescan sonar has become a common tool for seafloor surveys. They produce quality images of the seabed and a high probability of detection can be achieved when seeking specific underwater targets [10]. Autonomous underwater vehicle (AUV) systems have been viewed as offering a suitable platform for performing sea floor surveys using the sidescan sonar [11].

This work addresses the coverage problem that arises in subsea applications, and focuses on the offline algorithms that computes an optimal line sweep path for underwater coverage. This work is inspired by the applications in which the sensor only works well when it is traveling in a straight line, such as the subsea survey applications using sidescan sonar. Sidescan sonar sensor data is meaningful only when it is moving in a straight line [12], and it cannot be used in coverage algorithms that allow frequent turning maneuvers inside the cells. This work builds upon [13] which computes the optimal sweep direction of cells and the optimal cell decomposition for the coverage region consisting of polygonal cells.

There are extensive research results on the topic of robotic coverage in a known environment. For example, Moravec and Elfes [14] proposes an approximate cellular decomposition model, where the workspace is decomposed into cells with the same size and shape. Arkin and Hassin [15] proposes an approach based on covering salesman problem, which is a variant of the traveling salesman problem. Hert et al. [7] and Jung et al. [18] describes an algorithm for nonpolygonal region. Choset and Pignon [16] describes an offline coverage algorithm for polygonal region which performs line sweep decomposition called boustrophedon decomposition and creates a sequence of subregions using a heuristic Traveling Salesman algorithm. Schmidt and Hofner [17] describes a floor cleaning robot which has nonholonomic constraints. They use an offline coverage algorithm to generate a coverage path based on line sweep decomposition. A survey of papers on coverage for robotics can be found in [19]. Yang and Luo [20] presents a neural network approach for coverage with obstacle avoidance. Fang and Anstee [21] decomposes the surveyable area into subareas using an approximation to the generalized Voronoi diagram and calculate the sub-area paths to obtain a mission plan. Works that address the optimal coverage have also been published extensively. For example, Arkin et al. [22] shows that coverage problems that minimize the number of turns executed and the travel distance on planar rectilinear regions are NP complete in general. Gabriely and Rimon [23] considers the problem of covering a continuous planar area by a square-shaped tool attached to a mobile robot. Jinemez [24] presents a coverage planning based on genetic algorithms. Unfortunately, these optimal coverage algorithms that allow frequent turning motions inside the cells cannot be used in the underwater coverage mission using sidescan sonar, because the data obtained from the sidescan sonar is useful only in a straight line motion.

Of the approaches to coverage planning, line sweep based coverage algorithm such as the boustrophedon

† Corresponding Author: Division of Electrical and Electronic Engineering, Kangwon National University, Korea. (mhchoi@kangwon.ac.kr)
Received: June 6, 2012; Accepted: August 3, 2012

cellular decomposition ([16, 25]) is the most closely related with the way sidescan sonar is used in the seabed surveys. Huang [13] uses the boustrophedon approach to achieve the optimal coverage. Huang's approach seeks to minimize the number of turns required to cover all cells. This is consistent with minimizing the mission time. Huang shows that the optimal line sweep decomposition must use a sweep path that is parallel to an edge of the cell boundary, and computes the optimal cell decomposition of the coverage region and the optimal sweep directions of the cells. Huang's finding [13] motivates this research work which investigates the optimal line sweep coverage when the sweep path of a cell is parallel to an edge of the cell boundary. The optimal cell decomposition and the sweep directions of [13], however, do not lead to the optimal sweep path of the coverage region, as the cell sweep sequence is still unknown. It may be said that the optimal cell sweep sequence can be computed by solving a traveling salesman problem (TSP). However, in order to formulate this problem into a TSP problem, the exact location of sweep start and finish must be defined which determine the travel times between the cells. Furthermore, since the optimal sweep directions of each cell computed in [13] cannot be independent of the cell sweep sequence, the cell sweep sequence and the optimal sweep direction of each cell must be solved together. These important aspects of the problem have been left unexplored so far in the research community, and these are the main focuses of this paper.

In this paper, an optimal line sweep coverage solution is presented when the line sweep path is parallel to an edge of the cell boundary. The optimal sweep path of the cells, the optimal cell visit sequence, and the optimal cell decomposition of the coverage region are computed. The total time to complete the coverage task is minimized which is the sum of the sweep times of the cells and the travel times between the cells. The concept of sweep entry point and sweep exit point of a cell is introduced and used to compute the travel time between cells. The main contribution of this work is that the proposed optimal coverage solution computes the cell sweep sequence in addition to the optimal sweep paths inside the cells and the optimal cell decomposition. The optimal coverage problem is formulated as a group traveling salesman problem (Group TSP), which is converted to a binary integer programming problem and solved. To the best of author's knowledge, the cell visit sequence, the travel time between the cells and the sweep entry point and exit point have not been studied before in the optimal coverage problems in the research literature. The use of Group TSP in the solution of the optimal coverage problem is also an original contribution of this paper.

This paper is organized as follows. In Section 2, the formulation of the optimal coverage problem as a Group TSP problem and its solution procedures are described for a given cell decomposition. The optimal line sweep path and the cell visit sequence for a given cell decomposition is computed. In Section 3, the optimal line sweep path, the cell visit sequence and the optimal cell decomposition of the coverage region is computed. In Section 4, the proposed solution procedure is applied to a cell configuration example, and the optimal solution is analyzed, followed by discussions on the performance and the limitations of the proposed solution procedure in Section 5.

2. Optimal Line Sweep Path of a Cell Decomposition

2.1 Assumptions

Coverage region is assumed to consist of non-overlapping polygonal cells. This work uses the same cell configuration model of the coverage region as the one used in Huang [13], for easy and clear comparison of the simulation results with Huang's work. The coverage region consists of four cells and an island shown as shaded at the center of Fig. 1. The sweep path that covers each cell is composed of a sequence of parallel straight lines. The AUV executes a back and forth line sweep motion within each cell. Since the sidescan sonar data is valid only when the sonar is moving in a straight line, it is assumed that the turning maneuvers are done outside the cell to be covered. The narrow dead band region beneath the side scan sonar where there is no sensor data is not considered. This work considers the case where the line sweep path inside each cell is parallel to an edge of the cell boundary. Each cell is covered in a single continuous sweep as shown in Fig. 2(b), in a series of non-overlapping adjacent strips of equal width except for the last strip of the cell, where the width of the last strip may be less than the sweep width as shown by the shaded triangular region.

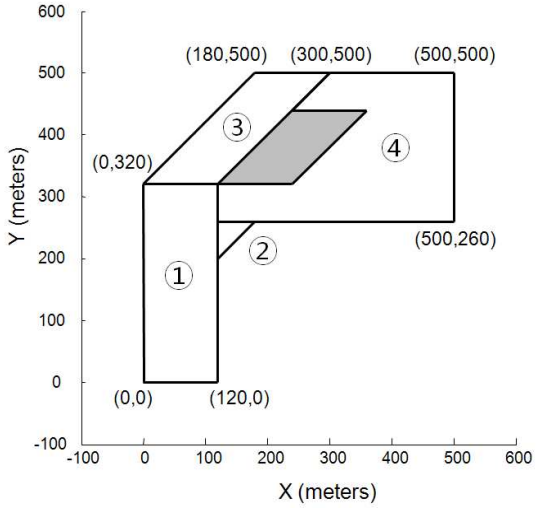
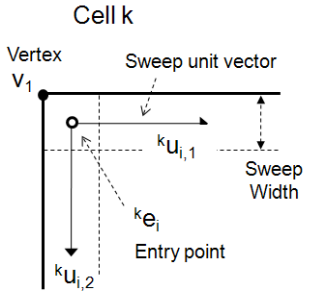
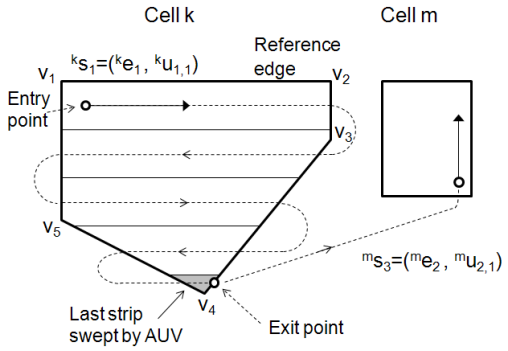


Fig.1. Cell configuration model of the coverage region



(a) Definition of entry point and sweep unit vector



(b) Definition of exit point and an example of line sweep

Fig. 2. Entry point, exit point and an example of cell sweep

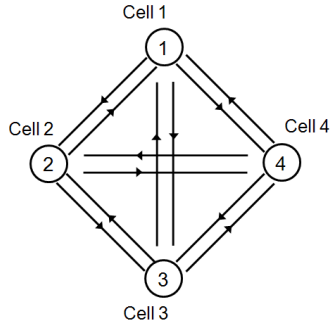
2.2 Formulation of the Optimal Line Sweep Path Problem as a Group TSP

The cell configuration model of the coverage region in Fig. 1 can result in a number of cell decompositions, varying from a decomposition consisting of four individual cells as shown in Fig. 1 to a decomposition consisting of a single cell with all four cells merged together. This work differentiates the two optimal line sweep paths: 1) the optimal line sweep path of a given cell decomposition and 2) the optimal line sweep path of the coverage region. The former is computed when each cell is swept individually. The latter is computed from all possible cell decompositions that can result from merging the cells in the cell configuration. A two step approach is used to solve the optimal line sweep coverage problem. In step 1, the optimal line sweep path of a given cell decomposition is computed. Objective cost is the total coverage time which is the sum of cell sweep times and travel times between cells. In step 2, the optimal line sweep path of the coverage region is computed. Since merging cells into a larger cell can result in a shorter mission time, all possible cell merge possibilities, i.e. all possible cell decompositions, are investigated. The mission time and the optimal sweep path of each cell decomposition are computed using the solution procedure in step 1. The cell decomposition resulting in the minimum mission time is selected as the optimal cell decomposition of the coverage region and the corresponding sweep path as the optimal sweep path. The procedure for the step 1 is described in the rest of this section, while the step 2 is illustrated in Section 3.

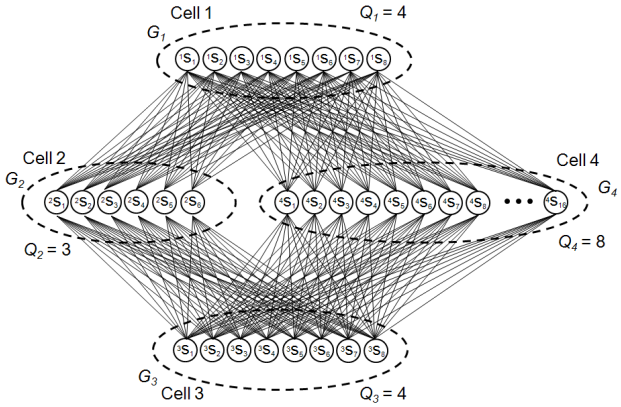
Problem1: Compute the optimal line sweep path when a cell decomposition consisting of polygonal cells is given and each cell is swept individually. Objective cost is the sum of cell sweep times and travel times between cells.

The solution procedure is illustrated using the cell decomposition shown in Fig. 1. The sweep path of AUV is defined as the path of the sensor position such as the sidescan sonar. In order to compute the travel time between cells, the concepts of entry point and exit point of a cell are introduced. Let the vertices of the cell k with Q_k vertices be denoted as v_m, m=1,…, Q_k. Sweep path of a cell begins near a vertex of the polygon. There exist two possible direction vectors for the sweep path originating from each vertex, and the initial sweep paths are located inside the polygon boundary by one half of the sweep width as shown in Fig. 2(a). The intersection of the two possible sweep paths is defined as the *entry point* of the vertex, and the entry points of cell *k* with Q_k vertices are denoted as $^{k}e_i$, i=1,…,Q_k. Let the edge of the cell boundary to which the sweep path is adjacent and parallel be called a *reference edge*. In Fig. 2(b), the reference edge is the line segment v_1-v_2. Let the unit sweep direction vector be called *sweep unit vector* and the two possible sweep unit vectors for the *i*-th entry point of cell k, $^{k}e_i$, be denoted as $^{k}u_{i,1}$ and $^{k}u_{i,2}$. *Sweep motion* is defined as a pair of an entry point and a sweep unit vector such that the set of sweep motions for cell *k* can be written as $\{^{k}s_n$, n=1, …, $2Q_k\}$ = $\{(^{k}e_i, ^{k}u_{i,j})$, i=1,…,Q_k and j=1,2$\}$. The sweep path moves towards the vertex that is farthest from the reference edge. The farthest vertex in the example of Fig. 2(b) is v_4. A sweep motion, $^{k}s_n$, thus completely describes a sweep of a cell by specifying the entry point and the sweep unit vector. A cell with Q_k vertices needs to be covered with only one of $2Q_k$ possible sweep motions. The intersection of the sweep path of the last strip and the cell boundary is defined as an *exit point*, as shown in Fig. 2(b). Let us assume that the next cell to sweep is cell *m*. After the sweep of cell *k*, AUV travels from the exit point of cell *k* to an entry point of cell *m*. It is assumed that the AUV arrives at the entry point of cell *m* with the vehicle heading aligned with the sweep unit vector ready to begin sweeping.

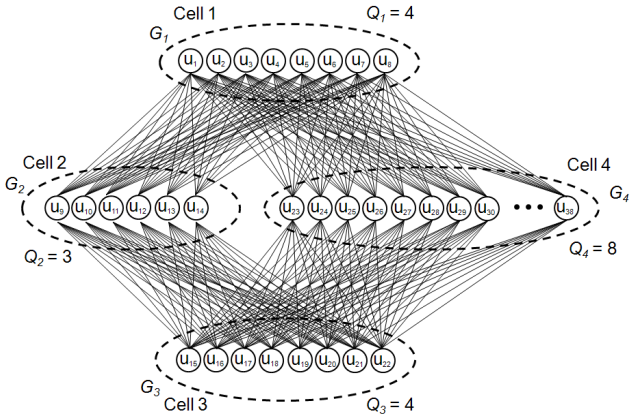
Connection graph of the cells is a graph where a node represents a cell, and a directed edge from cell *i* to cell *j* represents a visit from cell *i* to cell *j*. Tour of the cells is often represented by a connection graph. The connection graph for the cell model is shown in Fig. 3(a), where visits from a cell to all other cells are possible. In order to deal with the coverage of the cells in terms of entry points and sweep unit vectors, the concept of *coverage graph* is introduced, which is defined to be a graph where a node represents a sweep motion $^{k}s_i$, and a directed edge from node $^{k}s_i$ to node $^{m}s_j$ represents the combination of the



(a) Connection graph



(b) Coverage graph



(c) Coverage graph with node numbers u_i

Fig. 3. For ease of readability, two directed edges between a pair of nodes are drawn as an undirected edge between the two nodes. The edges between cell 1 and cell 3, and the edges between cell 2 and cell 4 are not shown in the graph (b) and (c).

sweep of cell k with the sweep motion ${}^{k}s_i$ and the travel from the exit point of cell k to the entry point of ${}^{m}s_j$. The coverage graph is constructed from the connection graph by replacing the cell k of the connection graph with a group of sweep motions $\{{}^{k}s_1, {}^{k}s_2, \ldots, {}^{k}s_{2Q_k}\}$ of the cell k as shown in Fig. 3(b). The sweep motions are the nodes of the coverage graph. This reflects the fact that visiting one of the sweep motions in the group for a cell is equivalent to visiting the cell. A directed edge from cell i to cell j in the connection graph is replaced by all possible directed edges from the sweep motions for cell i to the sweep motions for cell j in the coverage graph. The edge cost from node ${}^{k}s_i$ to node ${}^{m}s_j$ is defined to be the sum of the sweep cost of cell k with the sweep motion ${}^{k}s_i$, and the travel cost from the exit point of cell k to the entry point of ${}^{m}s_j$ arriving with the vehicle heading direction aligned with the sweep unit vector of ${}^{m}s_j$. By definition, the edge cost from node ${}^{k}s_i$ to node ${}^{m}s_j$ includes the sweep cost of cell k but does not include the sweep cost of cell m, and thus is not equal to the reverse edge cost, i.e. the edge cost from node ${}^{m}s_j$ to node ${}^{k}s_i$.The coverage graph is thus a directed graph. Edges between the nodes in the same group are not allowed since visiting two nodes in the same group would mean sweeping the same cell twice. For ease of readability, two directed edges between a pair of nodes are drawn as a single undirected edge in Fig. 3(b). For example, the undirected edge from ${}^{1}s_1$ to ${}^{2}s_1$ in fact represents two directed edges: a directed edge from ${}^{1}s_1$ to ${}^{2}s_1$ and a directed edge from ${}^{2}s_1$ to ${}^{1}s_1$. Also, the edges between nodes of cell 1 and nodes of cell 3, and the edges between nodes of cell 2 and nodes of cell 4 must be present but are not shown in Fig. 3(b) for ease of readability.

Let the number of cells be denoted as M, and the groups of nodes G_i corresponding to cell i be written as $G_i = \{{}^{i}s_1, {}^{i}s_2, \ldots, {}^{i}s_{2Q_i}\}$. Let V denote a set of all nodes of G_i, $i = 1, \ldots, M$, such that

$$V = \{{}^{1}s_1, {}^{1}s_2, \ldots, {}^{1}s_{2Q_1}, {}^{2}s_1, {}^{2}s_2, \ldots, {}^{2}s_{2Q_2}, \ldots, {}^{M}s_1, {}^{M}s_2, \ldots, {}^{M}s_{2Q_M}\}$$

The indices of the nodes in V need to be rearranged in a single sequence of numbers, so the set U is defined using V so that the elements of U and V are exactly the same, but the indices of U begins from 1 and ends at N, where N is the total number of nodes given by $N = \sum_{i=1}^{M} 2Q_i$.

$$U = \{u_1, u_2, \cdots, u_N\} \quad \text{where}$$
$$u_1 = {}^{1}s_1, \; u_2 = {}^{1}s_2, \ldots \; u_N = {}^{M}s_{2Q_M}$$

The element u_i is a node of the coverage graph. The node u_i describes a sweep motion of a cell by specifying the entry point and the sweep unit vector. The set of node indices belonging to cell k can be computed as follows.
$I_k = \{n | \; a_k \le n \le b_k\}$,

$$\text{where} \quad a_k = 1 + \sum_{i=1}^{k-1} 2Q_i \,, \; b_k = \sum_{i=1}^{k} 2Q_i$$

Then, the nodes u_i, $a_k \le i \le b_k$ in the set U belong to cell k. The coverage graph expressed using these node numbers is shown in Fig. 3(c). All possible visits between the nodes are included in the coverage graph. Hence, the coverage

graph represents all combinations of the cell sweep paths possible from the cell decomposition.

The optimal line sweep path of the given cell decomposition can now be solved by searching for the optimal tour of the coverage graph such that the tour passes through all groups $G_1, G_2, ..., G_M$, passing through only one node in each group, and return to the starting node. This is a problem referred to as a group traveling salesman problem (Group TSP). Since the Group TSP problem is known to be NP-hard ([26]), and it is an active area of research in algorithm research community (see for example, [27]), this work does not attempt to propose a general solution algorithm. Instead, the Group TSP problem is formulated as a binary integer programming problem, and the optimal solution is computed. Since the feasible solution space is well defined in the binary integer programming, the solution is found by exhaustive searching of the feasible solution space, and it is shown that in practice, a problem of a moderate scale can be solved quickly.

2.3 Formulation of the Group TSP as a binary integer programming problem

Binary integer programming formulation proposed by [28] for TSP problem is modified and used to formulate the Group TSP problem. Let

c_{ij} : cost of the edge from node u_i to node u_j
x_{ij} : binary integer variable corresponding to the edge from node u_i to node u_j

such that

x_{ij} = 1 if the edge is included in the tour
x_{ij} = 0 if the edge is not included in the tour

Then, the Group TSP problem is formulated as a binary integer programming problem as follows.

$$\text{Minimize} \qquad J = \sum_{i,j} c_{ij} x_{ij} \tag{1}$$

Subject to

$$x_{ij} \in \{0,1\} \quad i, j = 1, 2, \ldots, N \tag{2}$$

$$\sum_{i \in I_k} \sum_{j=1}^{N} x_{ij} = 1 \text{ , } k=1, \ldots, M \tag{3}$$

$$\sum_{j \in I_k} \sum_{i=1}^{N} x_{ij} = 1 \text{ , } k=1, \ldots, M \tag{4}$$

$$\sum_{i \in R} \sum_{j=\overline{R}} x_{ij} \geq 1, \text{ where}$$
$$R = \{n \mid a_k \leq n \leq b_k, \ \forall G_k \in S\},$$
$$S \subset \{G_1, G_2, \cdots, G_M\}$$
$$\text{such that } 2 \leq |S| \leq M-2 \tag{5}$$

Cost function in (1) is the sum of all edge costs included in the tour. Constraint (2) is the binary integer variable constraint. Constraint (3) specifies that number of edges going out from each group of nodes is equal to one, and (4) specifies that number of edges going into each group of nodes is equal to one. Constraint (5) is the subtour elimination constraint and specifies that no subset of the groups consisting of two or more groups is isolated from other groups. This constraint is used to avoid two isolated tour paths visiting all groups in the graph. In (5), S denotes a subset of the set of all groups $\{G_1, G_2, \cdots, G_M\}$, such that it contains two or more and M-2 or less groups, and R denotes the set of all node indices belonging to the groups in S. For example, if M=4 and $S = \{G_2, G_4\}$, then $R = \{a_2, a_2+1, ..., b_2, a_4, a_4+1, ..., b_4\}$, $\overline{R} = \{a_1, a_1+1, ..., b_1, a_3, a_3+1, ..., b_3\}$, and $|S|$ denotes cardinal number of the set S, which is equal to the number of groups in S.

The problem formulated in (1)-(5) is a binary integer programming problem and can be solved using integer programming techniques such as MATLAB bintprog.m function. However, the computational load can be prohibitively large. For the example cell configuration in Fig. 1, the number of cells M=4 , number of nodes N = 8+8+16+6 = 38, and the number of variables x_{ij} is $N^2 = 38^2$ = 1444. The value of each of the variables can be either 0 or 1, so finding the solution of the Group TSP is equivalent to searching for the optimal solution in the solution space of 2^{1444} vectors which is equivalent to infinite. The attempt to utilize the bintprog.m in MATLAB proved to be computationally infeasible using Windows 7, Intel Core i5 2.8GHz CPU machine with 4GB memory.

Even though the size of the solution space of Group TSP is extremely large, the size of the feasible solution space of x_{ij} satisfying (2)-(5) is relatively small, and can be used to solve the Group TSP problem. Let the set {1,2,3,4} represent the set of group numbers. The number of sequences by which the groups can be visited is $P(M, M)= M!=4!=24$, where $P(M, M)$ denotes the number of M-permutations of a set of M elements. The group visit sequences in this example are given by 4-permutations of the set {1,2,3,4}. For each of the group visit sequences, there are $\prod_{i=1}^{M} 2Q_i$ = 8*8*16*6 = 6144 different ways the nodes can be visited. The total number of node visit sequences in the coverage graph is thus N_e = 24*6144 = 147,456. The expression for this number in general is given by $N_e = M! \prod_{i=1}^{M} 2Q_i$. For each of these node visit sequences, a vector X_k={ x_{ij}} is assigned, so that element x_{ij} that has a value 1 implies the corresponding edge is included in the node visit. Since X_k, k=1,…, N_e is the set of all possible tours of the coverage graph satisfying (2)-(5), the remaining task in solving the Group TSP problem is to search for the X_k that results in the minimum cost J in (1). This method requires exhaustive search of N_e=147,456 feasible solution vectors and computationally feasible while direct application of binary integer programming to Group TSP problem requires binary search of 2^{1444} vectors.

The Problem 1 is formalized below for the convenience of future reference.

$$(J^*, X^*) = \textbf{Problem1}\ (D) \tag{6}$$

where the optimal sweep path $X^*=\{ x^*_{ij}\}$ results in the minimum mission time $J^* = \sum_{i,j} c_{ij}x_{ij}$ given the cell decomposition D = {Cell 1, Cell 2, …, Cell M}

3. Optimal Line Sweep Path of the Coverage Region and the Optimal Cell Decomposition

In Section 2, the optimal line sweep path of the given cell decomposition and the mission time were computed. Since merging cells into a larger cell can result in a shorter mission time, the mission times and the sweep paths of all possible cell merges, i.e. all cell decompositions, are computed using **Problem 1**. Huang [13] used merging multiple cells into a single cell using adjacency graph and dynamic programming. This paper examines the partitions resulting from the given cell configuration.

The total number of partitions of an n-element set is given by the Bell number B_n ([29]) which is defined by

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k \ , \ B_0 = 1,\ B_1 = 1$$

The first several Bell numbers are $B_0 = 1$, $B_1 = 1$, $B_2 = 2$, $B_3 = 5$, $B_4 = 15$. The list of all partitions, $P_i, i = 1,\ldots, B_M$, of a set of M cells can be computed by the script SetPartition.m obtained from the MATLAB Central webpage ([30]). All partitions of the set of four cells {1,2,3,4} are given by

P_1={1,2,3,4}, P_2={1,2,3}+{4},
P_3={1,2,4}+{3}, P_4={1,2}+{3,4},
P_5={1,2}+{3}+{4}, P_6={1,3,4}+{2},
P_7={1,3}+{2,4}, P_8={1,3}+{2}+{4},
P_9={1,4}+{2,3}, P_{10}={1}+{2,3,4},
P_{11}={1}+{2,3}+{4}, P_{12}={1,4}+{2}+{3},
P_{13}={1}+{2,4}+{3}, P_{14}={1}+{2}+{3,4},
P_{15}={1}+{2}+{3}+{4}.

For example, the partition P_2 = {1,2,3} + {4} consists of a large cell formed by merging cells 1,2 and 3, and a second cell which is the cell 4. The partition P_1 = {1,2,3,4} consists of a single cell obtained by merging the four cells 1,2,3 and 4, and the partition P_{15} ={1}+{2}+{3} +{4} is the partition consisting of the four individual cells. The partitions can be used as the cell decomposition input of Problem1. For each partition P_i, the optimal mission time J_i^* and sweep path X_i^* are computed, such that (J_i^*, X_i^*) = **Problem1**(P_i), $i = 1,\ldots,B_M$. The partition resulting in the minimum mission time is chosen as the optimal cell decomposition $\hat{P}^*$ and the corresponding optimal sweep path is selected as the optimal sweep path $\hat{X}^*$ of the coverage region. The solution algorithm for **Problem2** is summarized below.

Problem2: Compute the optimal cell decomposition $\hat{P}^*$ and the optimal sweep path $\hat{X}^*$ of the coverage region when the cell configuration is given.

Step 1: Compute all possible partitions of the cells, P_i, $i=1,\ldots, B_M$.

Step 2: Solve **Problem1** using each of the partition P_i as the cell decomposition input.
(J_i^*, X_i^*) = **Problem1**(P_i), $i = 1,\ldots, B_M$

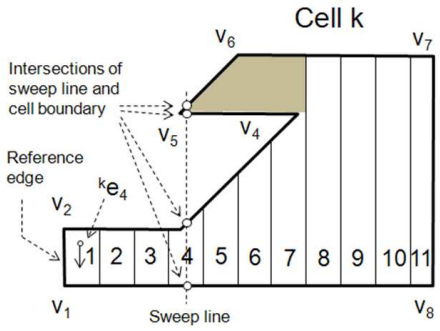
Step 3: Find the optimal solution
$(\hat{X}^*, \hat{P}^*) = (X_m^*, P_m)$ such that
$m = \arg_i \min\{J_i^*\}, i = 1,\cdots, B_M$

4. Simulation Results

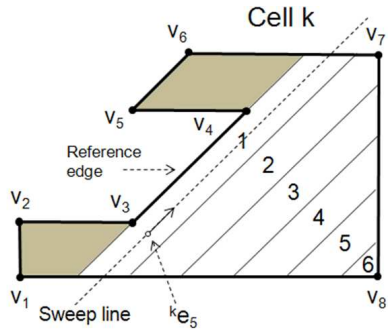
The cell configuration used in the simulation is shown in Fig. 1, where the coverage region consists of four cells and an island, and is the same as the model used in [13]. MATLAB was used to implement the proposed algorithm on a computer with Windows 7 on Intel Core i5 CPU, 2.8GHz machine with 4GB memory. Parameters used in the simulation are as follows. Sweep width of 15 meters was used initially, and then varied to 18 and 20 meters. The AUV is moving at the linear speed of three knots. For simplicity, each turn between the parallel sweeps is assumed to consist of a half circle of diameter equal to the sweep width at the speed of 1/3 of the linear speed. The travel time from the exit point of a cell to the entry point of the next cell was computed from the linear path from the exit point to the entry point. It is noted that these simplifications are for illustration purposes only, and these costs can be computed to any desired degree of accuracy from the geometry of the cells and the motion capability of the AUV, and used in the same framework of the proposed solution procedures.

A cell boundary is treated as a set of line segments between the vertices. A cell consisting of k vertices is represented by a set of coordinates of vertices, $\{(x_1,y_1), (x_2,y_2), \ldots, (x_k,y_k)\}$, such that the vertices are numbered beginning from an arbitrary vertex and proceeding along the polygon in a clockwise direction. For a given sweep motion, the interior of a cell can be divided into a series of strips of equal sweep width using the entry point and the sweep unit vector. The AUV moves along the sweep line which is the center line of the strip. The sweep distance and the resulting sweep time can be computed from the line equations of the sweep line and the line segments of the polygon. By the assumptions explained in Section 2.1, if the cell sweep cannot be completed in a series of adjacent strips, the sweep motion is said to be invalid, and the corresponding sweep time is set to be infinite. There can be two cases of invalid sweep motion: (a) If the sweep line of

a strip intersects the line segments of the polygon at more than two points, then the cell sweep cannot be completed in a series of adjacent strips. An example is shown in Fig. 4(a) where the sweep line of strip 4 intersects the cell boundary at four points. The same is true for strips 5, 6 and 7. In this case, when the sweep motion comes to an end at strip 11, the shaded part of the cell will remain unswept. This case can be detected by computing the intersection points of the line equations of the sweep lines and the line segments of the cell boundary. (b) If the series of adjacent strips generated by a sweep motion comes to an end while a part of the cell remains unswept, then the sweep cannot be completed in a series of adjacent strips. An example is shown in Fig. 4(b) where the sweep motion beginning from the entry point ${}^{k}e_5$ sweeps from strip 1 to strip 6, leaving the shaded parts of the cell unswept. This case can be detected by examining the relative positions of the vertices with respect to the reference edge. In the example of Fig. 4(b), the reference edge is the line segment v_3-v_4. If the vertices of the cell are on both the left and right sides of the segment v_3-v_4, the sweep motion is invalid. The vertices v_1, v_2, v_5 and v_6 are on the left side while vertices v_7 and v_8 are on the right side of the reference edge. These two types of invalid sweep motions are not possible in a convex polygon, but can occur in a non-convex polygon as shown in Fig. 4.



(a) Invalid sweep motion case 1



(b) Invalid sweep motion case 2

Fig. 4. Invalid sweep motions that cannot be completed in a series of adjacent strips. The shaded area remains unswept after the sweep of the cell is completed.

The optimal solution of **Problem1** for the partition P_{15} is shown in Fig. 5. The optimal sweep sequence is found to be {1} – {2} – {4} – {3}. The optimal tour by AUV begins from cell 1 at the entry point marked by a small circle at the top left corner of cell 1. The entry point is shown magnified in the figure. Turning paths are not shown. The sweep path of cell 1 ends at the exit point marked by a small triangle at the top right corner of cell 1. The entry point of a cell is marked with a small circle and the cell number, while the exit point is marked with a small triangle and the cell number. The AUV travels from the exit point of cell 1 to the entry point of cell 2 marked by a small circle at the top right corner of cell 2 along the path marked by a dotted line. lower left corner of cell 4. Sweep of cell 4 ends at the exit point at the top left corner of the cell, and travels to the entry point at the lower right corner of cell 3. After the sweep of cell 2 ends at the exit point at the top left corner of cell 2, the AUV travels to the entry point at the lower left corner of cell 4. Sweep of cell 4 ends at the exit point at the top left corner of the cell, and travels to the entry point at the lower right corner of cell 3. Sweep of cell 3 ends at the exit point at the lower left corner of the cell, and the AUV returns to the starting point of the tour which is adjacent to the exit point of cell 3. Since a travel from a cell to the next cell begins from an exit point marked by a triangle to an entry point marked by a circle, the closer the exit and entry points are, the smaller the travel distance, and they are often located adjacent to each other as can be seen in the figure. The cost of the entire tour consisting of the cell sweep time and the travel time between the cells was computed to be 7083.48 seconds.

A heuristic cell merge algorithm was developed and used in **Problem 2** to merge two cells into a single large cell. This algorithm can be applied repeatedly to merge two or more cells into one. Let us suppose cell A and cell B are to

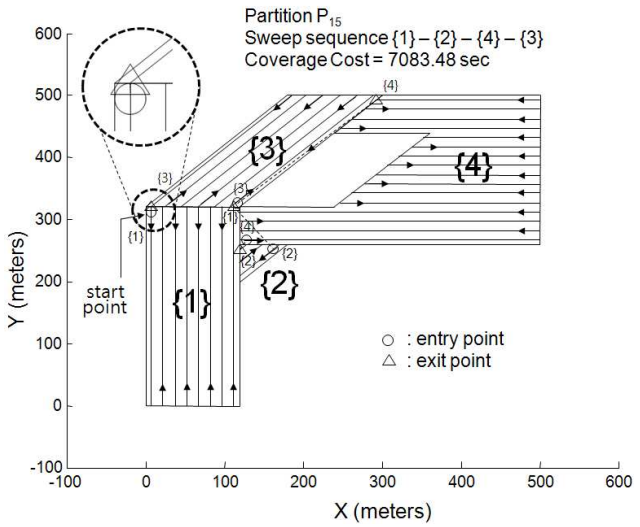
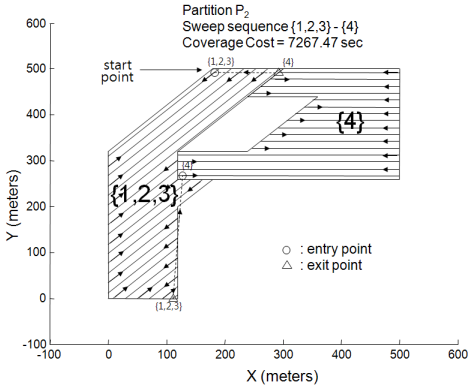
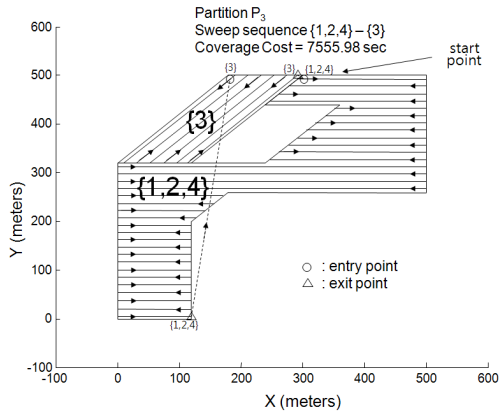


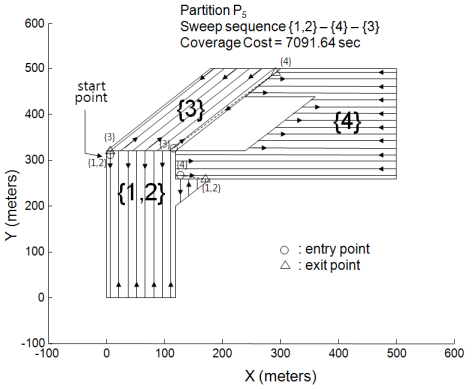
Fig. 5. Optimal sweep path of the partition P_{15}. (sweep width=15 meter). A small circle represents an entry point and a small triangle represents an exit point.



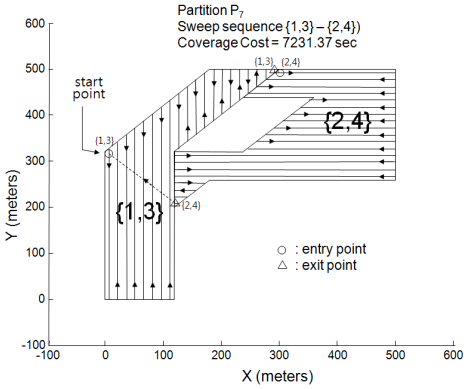
(a) Optimal sweep path of the partition P_2.



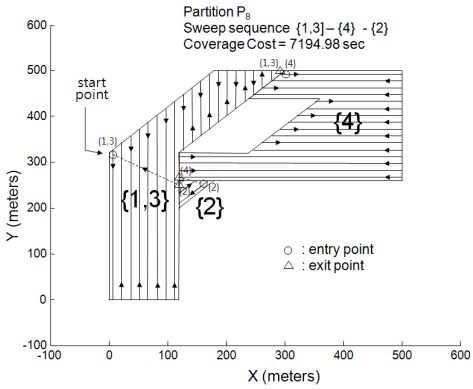
(b) Optimal sweep path of the partition P_3.



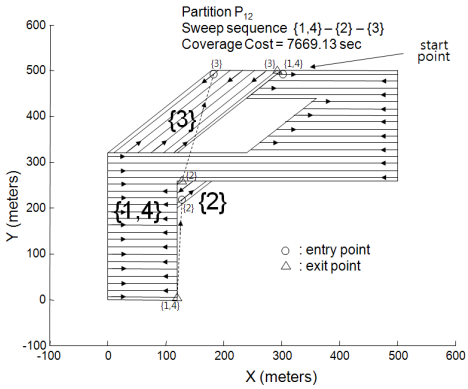
(c) Optimal sweep path of the partition P_5.



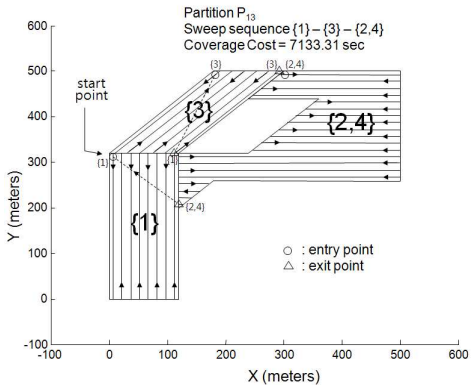
(d) Optimal sweep path of the partition P_7.



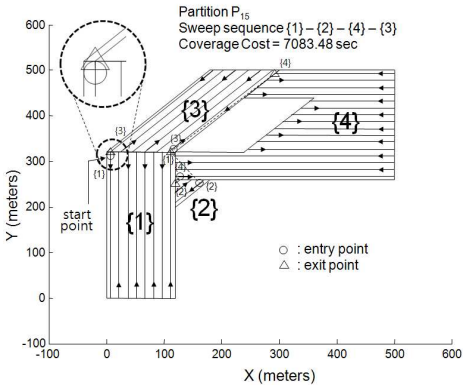
(e) Optimal sweep path of the partition P_8.



(f) Optimal sweep path of the partition P_{12}

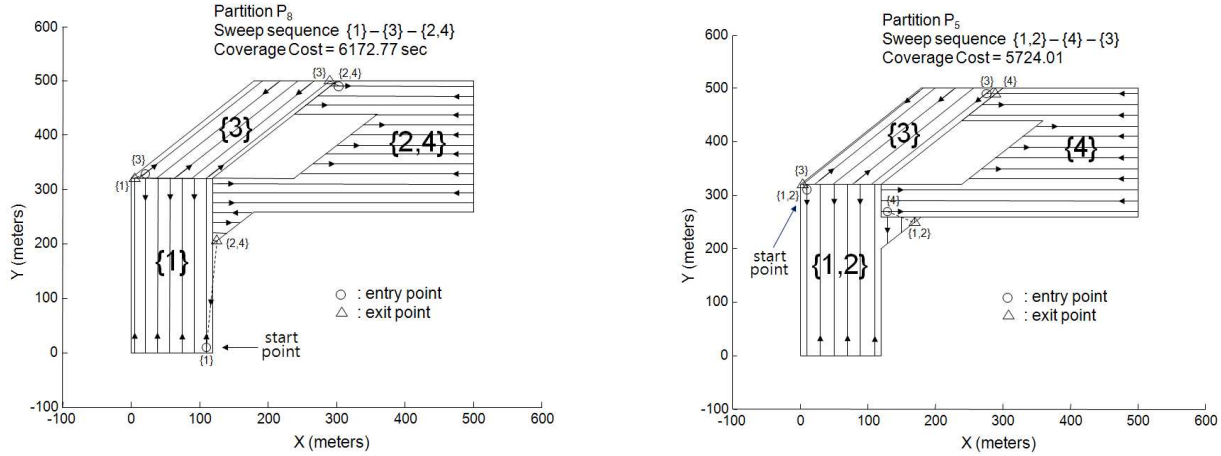


(g) Optimal sweep path of the partition P_{13}.



(h) Optimal sweep path of the partition P_{15}.

Fig. 6. Optimal sweep paths of the eight valid partitions (sweep width = 15 meter). The partition P_{15} is the optimal cell decomposition and the optimal sweep path of P_{15} is the optimal sweep path of the given coverage region.



(a) Sweep width = 18 meter

(b) Sweep width = 20 meter

Fig. 7. Optimal sweep paths of the coverage region for different sweep widths. Different sweep width results in different optimal cell decomposition and cell sweep sequence.

be merged. The cell merge algorithm examines the vertices of cell A that lie on the boundary of cell B and the vertices of cell B that lie on the boundary of cell A. Vertices of the merged cell are computed by removing the common line segment between the two cells. Merge of two cells fails if they do not share common segments as in the case of cell 2 and cell 3, or if the merging of two cells creates an island as in the case of cell 3 and cell 4. If all of the cell merges required in a partition can be completed, the partition is said to be valid. Otherwise, the partition is said to be invalid and not considered in the optimal solution search. Programming details are messy and thus omitted since it can be implemented in many different ways, and has relatively low technical importance.

Out of the 15 possible partitions, eight partitions P_2, P_3, P_5, P_7, P_8, P_{12}, P_{13}, and P_{15} are found to be valid, and the optimal solutions computed by **Problem 2** for the eight valid partitions are shown in Fig. 6(a) ~ (h). The sweep of each cell begins from the entry point marked by a small circle, and ends at the exit point marked by a small triangle. The sweep sequence and the sweep cost of each partition are shown at the top of the figures. The optimal line sweep path of the coverage region computed from the eight partitions is found to be the one from the partition P_{15} and is shown in Fig. 6(h), and Fig. 5. The optimal cell decomposition and the line sweep path of the coverage region shown in Fig. 6(h) agree exactly with the results of [13]. Additional information obtained in this work are the entry point and exit point of each cell, the sweep unit vectors from entry points, and the visit sequence of the cells.

The optimal partition and the optimal sweep path depend on many factors such as cell sizes, distances between cells, cost of sweep and turns, the travel time between cells, and the sweep width. For example, if the sweep width is changed, the number of strips covered in each cell can change, resulting in different locations of exit points. Then, the travel distances between cells change subsequently, and the optimal solution would have to be recomputed. For example, if the sweep width in Fig. 5 were changed from 15 to 18 meters, the number of strips swept in cell 1 would change from 8 to 7, and the exit point would be relocated from the top right corner to the lower right corner, resulting in the increased travel distance to cell 2. The optimal sweep paths of the coverage region computed by **Problem 2** for sweep widths of 18 and 20 meters are shown in Fig.7. For the sweep width of 18 meters, the partition P_8, where cell 2 and cell 4 were merged into a single cell, was found to be the optimal cell decomposition as shown in Fig. 7(a), while for the sweep width of 20 meters, the partition P_8, where cell 1 and cell 2 were merged into a single cell, was found to be the optimal cell decomposition as shown in Fig. 7(b). Comparing the optimal solutions in Fig. 7 with Fig. 5, it can be seen that the change in sweep width can result in the change in the optimal cell decomposition and the optimal sweep path.

5. Discussions

The total running time to compute the optimal solution of **Problem 2** in Fig.6 was 44.6 seconds on a computer with Windows 7 on Intel Core i5 CPU, 2.8GHz machine with 4GB memory. The running time for the optimal solution of **Problem 1** varies from 1.11 seconds to 32.9 seconds depending on the partition, with the partition P_{15} requiring the longest time.

The solution of the Group TSP problem was obtained by formulating it as a binary integer programming problem and exhaustively searching the feasible solution space. The number of feasible solutions satisfying (2)-(5) is given by $N_e = M! \prod_{i=1}^{M} 2Q_i$ where M is the number of cells and Q_i

is the number of vertices in cell *i*. The performance of the proposed solution will be ultimately limited by these factors.

Total number of partitions of an n-element set is given by the Bell number B_n, and the first few Bell numbers are: 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975. It is recommended by the author Luong [30] of the MATLAB script SetPartition.m that the number of elements, n, be kept less than or equal to 11 because of the recursive nature of their algorithm. This difficulty can be avoided if an iterative algorithm can be used to compute the partitions.

The procedures described in Section 3 are concerned with the algorithms for computing the optimal solution, and do not depend on how the sweep time and travel time between the cells are computed. These time costs were computed using simplifying assumptions in Section 4, but they can be computed with any desired degree of accuracy and be used in the same framework of the proposed solution procedures. The solution computed by the **Problem1** and **Problem2** in this work is optimal with respect to the assumptions made in Section 2.1.

Comparison of the major contribution of this work with the other existing works is summarized as follows. Optimal coverage solution presented in the literatures such as [22], [23], and [24] cannot be used in the underwater coverage mission using sidescan sonar, because they allow turning motions inside the cells while the data obtained from the sidescan sonar is useful only in a straight line motion. The straight line motion based coverage algorithms such as the boustrophedon cellular decomposition [16] are suitable for the seabed surveys using sidescan sonar. Among the straight line motion based coverage algorithms, Huang [13] is the only paper that studies the optimal coverage solution. Huang [13] presents an algorithm that computes the optimal cell decomposition and the sweep directions of each cell. Its cell coverage path is incomplete in the sense that the cell sweep sequence is not computed and the travel time between cells is not included in the optimality criterion. On the other hand, the optimal coverage solution presented in this paper computes the optimal cell visit sequence in addition to the optimal cell decomposition and the optimal sweep direction of the cells. Thus, the complete coverage path is computed in this paper. In terms of the technique used to solve the problem, Huang [13] uses adjacency graph and dynamic programming, while this work uses the Group Traveling Salesman Problem and cell partitions that can be created from the cell configuration. The optimal cell decomposition and the cell sweep direction results presented in [13] are exactly the same as the results in Fig. 5 in this work when the sweep width is 15 meters. However, this paper shows that the optimal coverage solution depends on the sweep width and thus different sweep widths can result in different optimal solutions as shown in Fig.7. Hence the optimal coverage solution in [13] is a special case of the more general results presented in this work.

6. Conclusion

In this paper, an underwater planar covering problem is studied where sidescan sonar is used for underwater coverage, and the sweep paths of the cells must consist of straight lines. A procedure that computes the optimal line sweep path of a given cell decomposition is proposed that computes the cell sweep sequence in addition to the sweep path inside the cells. The total time to complete the coverage task is minimized which is the sum of the travel times between the cells and the sweep times of the cells. The optimal coverage problem is formulated as a group traveling salesman problem (Group TSP) by introducing the concepts of entry point, exit point, and coverage graph. The Group TSP problem is converted to a binary integer programming problem, which is then solved using exhaustive search of the feasible solution space. The partitions of the cells are examined to compute the optimal cell decompositions and the optimal line sweep path of the given coverage region. The proposed solution algorithm is applied to a cell configuration example, and it is shown that a problem of a moderate scale can be solved quickly.

Acknowledgements

This study is supported by the Research Grant from Kangwon National University. The authors would like to thank Professor Daniel Stilwell, Bradley Dept of Electrical and Computer Engineering, Virginia Tech, for his help in conducting this research work.

References

[1] Yasutomi, F., Takaoka, D., Yamada, M., and Tsukamoto, K., “Cleaning robot control”, IEEE Int. Conf. Robotics Automation, Philadelphia, PA, pp. 1839-1841, 1988.

[2] Oh, J.S., Choi, Y.H., Park, J.B., and Zheng, Y.F., “Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map”, IEEE Trans. Industrial Electronics, Vol. 51, No. 3, pp. 718-726, 2004.

[3] Huang, Y.Y., Cao, Z.L. and Hall, E.L., “Region filling operations for mobile robot using computer graphics”, Proc. of the IEEE Conference on Robotics and Automation, pp. 1607-1614, 1986.

[4] Najjaran, H. and Kircanski, N., “Path planning for a terrain scanner robot”, Proc. 31st Int. Symp. Robotics, Montreal, QC, Canada, pp. 132-137, 2000.

[5] Ollis, M. and Stentz, A., “First results in vision-based crop line tracking”, Proc. IEEE Int. Conf. Robotics Automation, Minneapolis, MN, pp. 951-956, 1996.

[6] Farsi, M., Ratcliff, K., Johnson, P.J., Allen, C.R., Karam, K.Z., and Pawson, R., “Robot control system

for window cleaning", Proc. 11th Int. Symp. Automation Robotics Construction, Brighton, U.K., pp. 617-623, 1994.
[7] Hert, S., Tiwari, S., and Lumelsky, V., "A terrain-covering algorithm for an AUV", Autonomous Robots, Vol. 3, pp. 91-119, 1996.
[8] Reed, S., Petillot, Y., and Bell, J., "An Automatic Approach to the Detection and Extraction of Mine Features in Sidescan Sonar", IEEE J. of Oceanic Eng., Vol. 28, No. 1, pp. 90-105, 2003.
[9] Sariel, S., Balch, T. and Erdogan, N., "Naval Mine Countermeasure Missions", IEEE Robotics and Automation Magazine, March, pp. 45-52, 2008.
[10] Couillard, M., Fawcett, J. A., Myers, V.L., Davison, M., "Support vector machines for classification of underwater targets in sidescan sonar imagery", Technical Memorandum DRDC Atlantic TM 2008-190, November, 2008.
[11] Evans, B., Baralli, F., Bellettini, A., Bovio, E., Coiras, E., Davies, G., Groen, J., Myers, V., Pinto, M., "AUV technology for shallow water MCM recon-naissance", NURC Reprint Series, NURC-PR-2007-008, 2007.
[12] Paull, L., Saeedi, S., Li, H., Myers, V., "An Information Gain Based Adaptive Path Planning Method for an Autonomous Underwater Vehicle using Sidescan Sonar", 6th annual IEEE Conf. on Automation Science and Engineering, Toronto, Canada, pp. 835-840, 2010.
[13] Huang, W.H., "Optimal line-sweep-based decompositions for coverage algorithms", 2001 IEEE Conf. on Robotics and Automation, pp. 27-32, 2001.
[14] Moravec, H.P. and Elfes, A., "High resolution maps from wide angle sonar", IEEE Int. Conf. Robotics Automation, St. Louis, MO, pp. 116-121, 1985.
[15] Arkin, E.M. and Hassin, R., "Approximation algorithms for the geometric covering salesman problem", Discrete Appl. Math., 55, pp. 197-218, 1994.
[16] Choset, H. and Pignon, P., "Coverage path planning: The boustrophedon cellular decomposition", Int. Conf. on Field and Service Robotics, Canberra, Australia, 1997.
[17] Schmidt, G. and Hofner, C., "An advanced planning and navigation approach for autonomous cleaning robot operations", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 2, pp. 1230-1235, 1998.
[18] Y.S. Jung, B.H. Lee, M.H. Choi, S.Y. Lee, K.W. Lee, "An Efficient Underwater Coverage Method for Multi-AUV with Sea Current Disturbance", Int. Journal of Control, Automation and Systems, Vol. 7, No. 4, pp. 615-629, Aug., 2009
[19] Choset, H., "Coverage for robotics - A survey of recent results", Annals of Mathematics and Artificial Intelligence, Vol. 31, pp. 113-126, 2001.
[20] Yang, S. and Luo, C., "A Neural Network Approach to Complete Coverage Path Planning", IEEE Trans. on Systems, Man, and Cybernatics - PART B, Vol. 34, No. 1, pp. 718-725, 2004.
[21] Fang, C. and Anstee, S., "Coverage Path Planning for Harbour Seabed Surveys using an Autonomous Underwater Vehicle", OCEANS 2010, pp. 1-8, 2010.
[22] Arkin, E.M., Bender, M. A., Demaine, E.D., Fekete, S.P., Mitchell, J.S.B. and Sethia, S., "Optimal covering tours with turn costs", Proc. of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, Washington. D, C., United States. pp. 138-147, 2001.
[23] Gabriely, Y. and Rimon, E., "Spanning-tree based coverage of continuous areas by a mobile robot", Annals of Mathematics and Artificial Intelligence Vol. 31, pp. 77-98, 2001.
[24] Jimenez, P. A., Shirinzadeh, B., Nicholson, A., Alici, G., "Optimal Area Covering using Genetic Algorithms", Proceedings of 2007 IEEE/ASME int. conf. on Advanced Intelligent Mechatronics, pp. 1-5, 2007.
[25] Garcia, E. and Gonzalez, P. de Santos, "Mobile-robot navigation with complete coverage of unstructured environments", Robotics and Autonomous Systems, Vol. 46, pp. 195-204, 2004.
[26] Safra, S and Schwartz, O., "On the Complexity of Approximating TSP with Neighborhoods and Related Problems", Computational Complexity Vol. 14, pp. 281- 307, 2005
[27] Elbassioni, K., Fishkin, A.V., Mustafa, N.H., and Sitters, R., "Approximation Algorithms for Euclidean Group TSP", 32nd International Colloquium, L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp. 1115-1126, 2005.
[28] Laporte, G.," The traveling salesman problem: An overview of exact and approximate algorithms", European Journal of Operational Research Vol. 59, pp. 231-247, 1992
[29] Rota, G., "The Number of Partitions of a Set", American Mathematical Monthly, Vol. 71, No. 5: pp. 498-504, 1964.
[30] Luong, B., "MATLAB Script for computing all partitions of a set of n elements", [webpage] http://www.mathworks.cn/matlabcentral/fileexchange/24133-set-partition/content/SetPartFolder/SetPartition. m., May, 2012

Myoung Hwan Choi He received B.S., M.S. and Ph.D. degrees in Electrical and Electronic Engineering from Seoul National University, Seoul, Korea in 1982, 1986, and 1992 respectively. He is a Professor in the Division of Electrical and Electronic Engineering, Kangwon National University, Chuncheon, South Korea. He was a visiting scholar in the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA in 2010. His research interests include robotic motion planning, multiple robot coordination and impedance control.