The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

3-D Terrain Covering and Map Building Algorithm for an AUV

Tae-Seok Lee, Jeong-Sik Choi, Jeong-Hee Lee, and Beom-Hee Lee, *Fellow, IEEE*

Abstract— **In order to improve the efficiency of a terrain covering algorithm for a robot(AUV), an artificial island(AI) technique is developed. Such an algorithm is necessary to acquire information and make a terrain map in an unknown three dimensional underwater environment. This algorithm considers that the three dimensional environment consists of a number of planes, at various ocean depths. The artificial island technique has the advantage of reducing the covering path length and the time cost for the robot, because it takes the correlation between every plane. In this paper, the concept of the artificial island technique is presented and its validity is proved under certain conditions. Through various simulations, we validate the efficiency in terms of the total path length and the running time of the AUV. An example of a complete three dimensional map obtained using this technique is provided.**

I. Introduction

Recently the underwater environment has become more important and has been researched. Acquiring information on the underwater environment is obviously necessary for works on topography, resource searching, and military applications. To achieve this goal, we propose the algorithm for the underwater terrain exploration.

In recent times, due to the improvement of sonars' resolution, drawing the map of the underwater surface using multi-beam sonars has become plausible. However, it cannot represent detail information of the underwater terrain. We can cope with this problem through covering the terrain near the underwater surface by using an autonomous underwater vehicle (AUV) [1]-[3]. In real situations, various commercial and military applications have been conducted successfully using an AUV for the deep sea. Through recent developments, AUVs can be employed at depth of up to 6000meters below the surface. For example, the REMUS 6000 has been developed for exploration in the deep sea as shown in Fig. 1. Therefore, the motion planning algorithm for an AUV has become important.

Several coverage applications use on-line coverage algorithms [3] [4], where the work space is unknown environment. Thus, these on-line coverage algorithms construct their motion trajectories step-by-step. Generally, these covering algorithms decompose the three dimensional space into two dimensional planes, with respect to the depth of the space, and the AUVs are operated in each plane. Oommen et al. developed a covering algorithm for an unknown planar terrain using visibility graph [5], and Rao et al. extended this algorithm [6]. However, these algorithms are limited to polygonal obstacles, so they are unsuitable for real environments. A more realistic algorithm with arbitrary planar shaped obstacles was in [7]. Hert et al. improved the algorithm in [7] by presenting a superior upper bound of the total path length of the AUV [8]. However, there is still room for improvement for the algorithm that covers several planes.

Fig. 1. Launching an AUV(REMUS 6000) [12].

Whether a single AUV or several AUVs are employed, in actuality, it is impossible to operate the AUV or AUVs so as to cover all of the planes of the underwater area simultaneously. Accordingly, the AUV has to cover a specific plane first, and then cover a nearby plane. Because of this sequential process, we inevitably acquire information from the covered plane before exploring other plane. However, most previous papers do not consider this prior information when they explore the three dimensional environment. Therefore, in this paper, we propose a method of incorporating the prior terrain information into the conventional coverage method. We compare the efficiency of the Hert algorithm, since it shows better results than others in terms of the total path length, with and without the proposed algorithm.

In [9], Herbert and Johnson have generated a map by using exploration robot's acquired sensor data. Various studies have been devoted to constructing a three dimensional underwater map [10] [11]. However, these algorithms only concentrate on the data association of acquired sensor data to generate an accurate underwater map. Differently, with the technique proposed in this paper, the acceleration of the map building process during data acquisition is improved.

In order to develop a framework for efficient underwater investigation, this paper proposes techniques for covering an

This work was supported in part by the Korea Science and Engineering Foundation(KOSEF) NRL Program grant funded by the Korea government(MEST)(No. R0A-2008-000-20004-0), the Growth Engine Technology Development Program funded by the Ministry of Knowledge Economy, Automation and Systems Research Institute(ASRI) in Seoul National University, and the Brain Korea 21 Project.

Tae-Seok Lee, Jeong-Sik Choi, Jeong-Hee Lee, and Beom-Hee Lee are with the School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul, Korea (e-mail: {felix84, jsforce, meeckee2, bhlee}@snu.ac.kr). B. H. Lee is an IEEE fellow and currently a professor in the School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul, Korea.

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

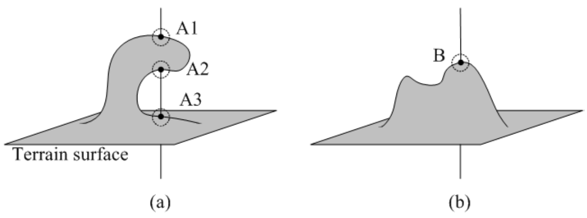


Fig. 2. Model of the underwater surface: (a) a non-vertically projective planar surface, (b) a vertically projective planar surface [8].

underwater terrain and building a three dimensional map. After describing the results of the various simulations, we discuss their validity.

II. Artificial Island Technique

A. AUV Modeling

In this paper, we modeled the AUV as a point, i.e. having no volume and weight, so it is considered as a holonomic vehicle. A fixed orthogonal coordinate system X=(x,y,z) is chosen with z axis passing through the center of the earth.

The AUV has range sensors for scanning and a camera for observing area underneath the AUV. The sensing region of this equipment forms a rectangular polyhedron with a dimension of l x w x h (length x width x height) with the AUV at the center of sensing region. Using the camera that is located on the lower part of the AUV, picture of rectangular space below the AUV with the bottom plane with dimensions of l x w with a depth of h/2 is taken.

B. Environment Modeling

The actual ocean floor consists of complex structures, so a continuous arbitrarily shaped surface model is required. The underwater terrain surface is categorized into two types. The first type of surface is a vertically projective planar surface similar to the one presented in Fig. 2(b). The vertically projective surface means that the topographic surface has a point intersected by a vertical line at any given position. On the other hand, non-vertically projective surface in Fig. 2(a) has three cross points on the surface by a particular vertical line. To develop a realistic algorithm, this paper considers both types of terrain.

However, there is still a restriction. We have to assume that the boundary surface is not curved as shown in Fig. 3. This indicates that the boundary does not bend such that it would deviate from the outer boundary by more than the AUV sensing range: w/2 or l/2. This is a realistic assumption, because in most cases, the real ocean surface bends only gradually on its upper surface, except in the caves. Since we are using the AUV to achieve a more accurate investigation near the boundary, we will not cover the empty region far from the boundary to curtail the path length of the AUV.

In the two dimensional projection on the xy-plane, the ocean surface is divided into a number of simple components as in Fig. 4. *Grid line* L_i is a vertical line on the xy-plane defined by the equation x=iw/2, where i is a non-negative integer. The interval between the grid lines varies according to the sensing width w. The AUV moves along the grid line and senses the area between the adjacent grid lines. *Bay* is a continuous concave part of the boundary which intersects the same grid line. Similarly, *cape* is a continuous and convex part of the boundary. *Island* is an area isolated from the outer boundary. *Inlet* is an area that has a cape point as its entrance and bay point as its end. The grid line segment at the entrance of the inlet is the *inlet doorway*. Since AUV has to pass through the entire segment to finish exploring, the performance and efficiency of the AUV are determined by the algorithm it uses.

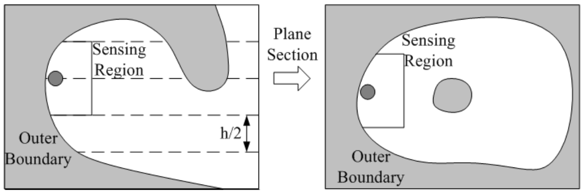


Fig. 3. Excluded shape type of the non-vertically projective surface.

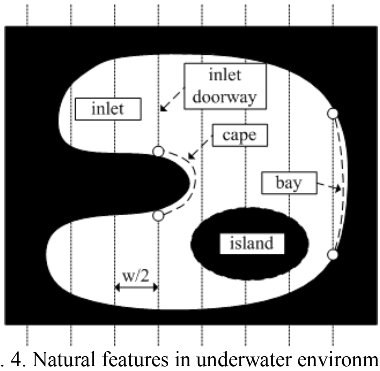
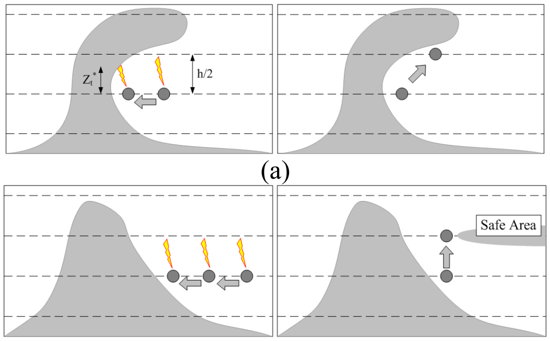


Fig. 4. Natural features in underwater environment.

C. General Terrain Covering Algorithm

In this part, the basis of the general terrain covering algorithm is explained concisely. As mentioned in section I, the Hert algorithm [8] is more efficient than the others, so we chose it as the criterion.

Most underwater covering algorithms, including the Hert algorithm, divide three dimensional space into two dimensional planes. Basically, the Hert algorithm operates on the plane in three steps; zigzagging, inlet covering, and island covering. The key to the Hert algorithm is a simple zigzagging route, where the AUV moves back and forth along the grid lines, sweeping across the area from either right or left. However, this simple zigzagging motion is not sufficient to cover complex boundaries. When the AUV senses an unexplored cape point during the exploration, it diverges to the cape point from the current grid line. After diverging to the cape point, the AUV starts to follow the boundary which contains the cape point. There are two cases which arise during boundary following. In the first case, when the AUV is in an inlet, it has to pass a bay during the boundary following. After passing the bay, by zigzagging, the AUV comes back to the cape point at which it diverged. The second case is where the AUV returns to the diverged cape point during the boundary following. This return means that the AUV have covered the connected boundary, an island. In this situation, the AUV moves from the diverged cape point to the next cape



(a)

(b)

Fig. 5. Illustration of the AUV sensing the boundary in upper plane: (a) the AUV works on the non-vertically projective surface, it senses the boundary within *h*/2, (b) the AUV works on the vertically projective surface.

point which is on the opposite side of the island. Then the AUV comes to the diverged cape point by zigzagging similar to inlet covering motion.

However, this algorithm disregards the relation between planes. In [8], how the AUV moves to a certain plane to an adjacent plane is explained. After relocating to new plane, the AUV runs original Hert algorithm again. Since it does not focus on the efficiency of the covering result in the new plane, it cannot guarantee the efficiency of the three dimensional terrain covering process.

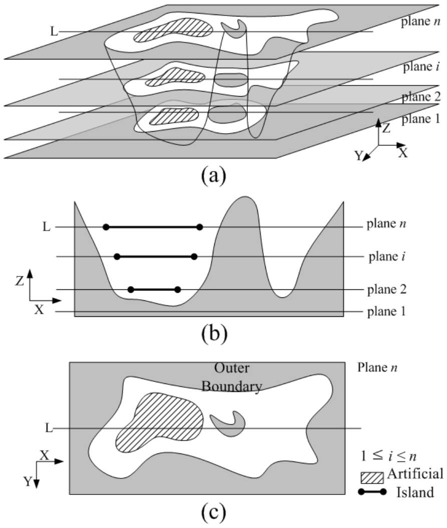
D. Analysis of the Artificial Island Technique

As stated above, common covering algorithms concentrate on the two dimensional covering method in three dimensional space. Therefore, by focusing on the efficiency of covering of a whole space rather than a plane, we can obtain better results in terms of the total path length or operating time cost.

As stated in section I, the AUV cannot cover the whole space at once, and should therefore cover the individual planes sequentially. Due to this sequential process, we can use the information obtained from the previously covered plane for the exploration of the other planes. However, most covering algorithms do not consider such technique.

Since most of the resources in the deep sea or mines are located near the surface boundary it is sufficient to sense whether the boundary is changing or not and conduct missions near it. Therefore, checking the change of the boundary is an important factor to operate the AUV efficiently. As mentioned in section II-A, the terrain surface is categorized into two types. The key to categorizing it is the slope of the boundary. As shown in Fig. 5(a), when the AUV is operating on the non-vertically projective surface, it can sense the changing of the boundary within the sensing range *h*/2 in the upper plane near the outer boundary. When the AUV is operating on the vertically projective surface, as in Fig. 5(b), it cannot sense the changing of the boundary.

If the range sensor perceives the boundary within *h*/2, the integrated area of the distribution under *h*/2, $p(z_t<h/2|x_t, m)$ becomes large. In contrast, when the sensor does not detect any obstacles within *h*/2, $p(z_t<h/2|x_t, m)$ becomes small. We define z_t as the measurement value, x_t as the robot pose at time *t*, *m* as the environment information, and η as the sensor reliability coefficient depending on the sensor. Then we can derive a new inequality (1). If the perception is larger than η, a boundary or obstacle exists in the upper plane with respect to the current position.



(a)

(b)

(c)

Fig. 6. Illustration of the artificial islands in three dimensional space: (a) three dimension view of the entire terrain, (b) a vertical sectional view of (a), (c) a cross sectional view of (a)

$$\eta \le p(z_t \le \frac{h}{2} \,|\, x_t, m) \tag{1}$$

Using (1), we can consider that, if there is an area without the boundary in the upper plane, unoccupied area need not be covered when the AUV is covering the upper plane. Because there are no obstacles or boundaries, the area is safe for the AUV. Therefore, it is defined as a safe area. We can skip this safe area, because we assume that the objects found exist near the boundary.

The main idea behind the *artificial island(AI) technique* is that the AUV covers the area excluding parts of the safe areas as shown in Fig. 6. This technique sets the parts of the safe area as AIs. Then, the AUV does not cover the AIs. When the AUV encounters the AIs during the covering, it considers them as general islands. As a result, the covering area of the AUV becomes *Covering Area = entire area – artificial island area*.

The AI technique is a dependent algorithm. This technique is an adaptive tool which reduces the total path length by applying with the conventional covering algorithms. Therefore, the strength of this technique is that it can be used with various covering algorithms.

To verify the efficiency of the adaptation of the AI, we compare the total path length of the AUV with and without this concept. Since the AI technique works with various covering algorithms, we first choose a suitable covering algorithm. As mentioned above, herein we compare and analyze the affect of the AI when it is adopted by the Hert algorithm. The total path length of the AUV, P_H, using the Hert algorithm is as follows [8]:

$$P_H \le L' + 3L'' + 2P' + 3P'' + 2Q' \tag{2}$$

where P_H is the total path length of the robot in the plane, P' is the length of the plane's outer boundary, and P'' is the sum of the lengths of the plane's island boundaries. Also, let L' be the sum of the lengths of all grid line segments, except for the inlet doorways, L'' the sum of the lengths of all inlet doorways, and Q' the sum of the lengths of all capes.

When AIs are deployed in the Hert algorithm, the total path length is increased by the increase in the boundary length of the AI and decreased because the grid line segments in the AI vanish. Thus, the sum of the lengths of all grid line segments and the sum of the lengths of the island boundaries becomes as follows:

$$L' \text{ of Hert's } \Rightarrow L' - L'_P \tag{3}$$

$$P'' \text{ of Hert's } \Rightarrow P'' + P''_P \tag{4}$$

where L'_P is the sum of the lengths of all grid line segments which overlaps with the AIs, and P''_P is the sum of the lengths of the boundaries of the AIs. When substituting L' and P'' in (2) with (3) and (4), the total path length, P_P, when the AI technique is adopted is expressed by (5):

$$P_P \le (L' - L'_P) + 3L'' + 2P' + 3(P'' + P''_P) + 2Q' \tag{5}$$

The difference between (2) and (5) is as follows:

$$P_H - P_P \le L'_P - 3P''_P \tag{6}$$

To make the left hand side of the inequality (6) larger than zero, the following condition should be satisfied.

$$3P''_P \le L'_P \tag{7}$$

According to this inequality, when (7) is satisfied, the AI technique contributes to shortening the path length compared to that of the original Hert algorithm.

When we modify the technique applied to the algorithm by Hert, more efficient results are acquired. The Hert algorithm is inefficient in its use of the island covering procedure, because the AUV has to travel one and half times the length of the boundary of the island to cover it completely. In the above, the AI is treated as a real island, but in fact, it does not have a real boundary. Also, the AI is a known environment different from the real island. Therefore, we do not need to explore one and half times the length of the boundary of the AI. The path length of the AUV can be reduced by adjusting this technique so that the AUV travels only half of the length of the AI boundary. Then, the extent of this increase, 3 P''_P is reduced to 2 P''_P. (5) and (7) are changed as follows:

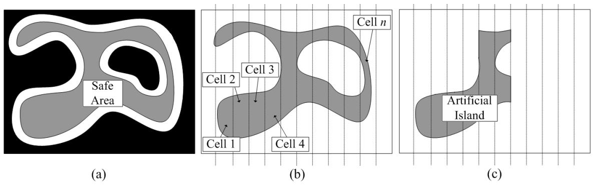


Fig. 7. Generating the artificial islands: (a) safe area after covering a particular plane, (b) dividing the safe area by the grid line width, (c) calculating the most reducible case and setting the artificial islands.

$$P_P \le (L' - L'_P) + 3L'' + 2P' + 3P'' + 2P''_P + 2Q' \tag{8}$$

$$2P''_P \le L'_P \tag{9}$$

The AI technique can provide more efficient results when a proper adjustment is applied according to the conventional covering algorithms.

The basic flow for the generation of the AI is as follows:

Step 1: A certain plane is covered by an ordinary planar covering algorithm.

Step 2: After the covering procedure is finished, the safe area is extracted from the plane as shown in Fig. 7(a).

Step 3: The AIs are calculated by arranging the Step 2 area.

Step 4: The upper plane is covered by referring to the AIs from Step 3.

Step 5: Go back to Step 2 until the procedure ends. The AIs in the former steps are included in the covered area.

Many methods can be used to generate the AI. The key procedure for generating the AI is Step 3. To make the AI technique more understandable, the following structure is applied to Step 3 in this paper:

Step 3-1: *Making the artificial island sets 1*: As shown in Fig. 7(b), the Step 2 area is divided into cells according to the grid line width.

Step 3-2: *Making the artificial island sets 2*: Using cell 1 to cell *n* from the Step 3-1, subsets are made. The AIs are generated using these subsets. The AIs that satisfy (9) are selected.

Step 3-3: *Deciding the final artificial islands*: When AIs overlap others, they are compared and finally the most efficient AI is chosen.

As stated above, other techniques can be used in place of this method. Adopting another method may affect the efficiency of the covering algorithm. However, the basic role of Step 3 does not change: making the artificial island sets, and deciding the final artificial islands.

III. Simulation Result

The AUV is modeled as a holonomic point vehicle and the dynamics of the vehicle or sea current is not considered in this paper. Therefore, the path length and the running time of the

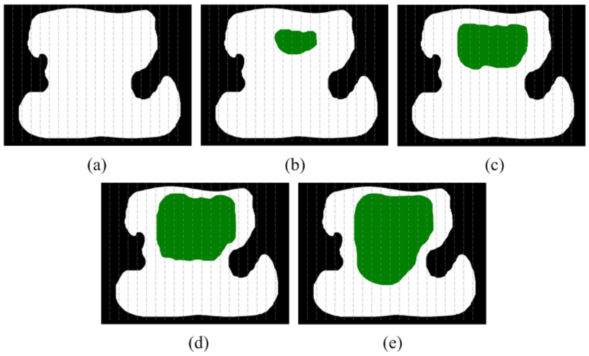


Fig. 8. Planar maps for simulation varying the area of artificial islands. The number matches TABLE I.

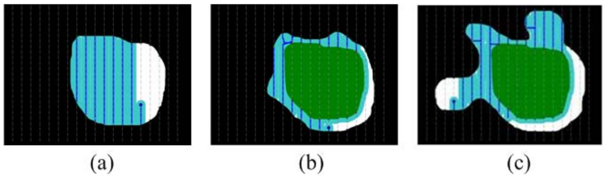


Fig. 9. Covering simulations varying the plane depth. The number matches TABLE II: (a) lowest plane, (b) middle plane, (c) highest plane.

AUV have directly proportional relation. The sensing width and length of the AUV are set to 40m and the sensing height is set to 100m. The velocity is 1.5m/s. The simulations are conducted in an area of 19,200m^2. Underwater marine life is not considered. All simulations are performed using MS Visual Studio 6.0.

A. *Simulation varying Artificial Island Area*

In order to check the relation between the efficiency and the area of the AI, a simulation is performed.

The experiment is conducted with an arbitrary outer boundary shape and performed again by applying different size AIs with the same outer boundary, as shown in Fig. 8.

The results are shown in TABLE I. Map (a) in Fig. 8 is the case with no AI. Maps (b) and (c) correspond to the case with a small size AI and maps (d) and (e) correspond to the case with a large size AI. The total path length increases slightly when (7) is used in the small AI situation. On the other hand it is always reduced when (9) is used. A detailed discussion is given in IV.

B. *Simulation as a Function of the Depth*

A simulation in three dimensional space was performed. As described in section II-B, the lower plane is covered first and then the AI is extracted. Following this, this sequence is reiterated. The simulation results are presented in Fig. 9 and TABLE II.

Since the total area increases as the depth increases, the path length and operating time of the AUV increase.

C. *Result of Covering Ulleung Basin Model*

This simulation is performed in a real model, the Ulleung Basin in the East Sea, Korea, at ocean depths between 1500m and 2400m. The simulation model is constructed based on ocean atlas map [13]. 10 planar maps are set at depth in intervals of 100m. The AUV started to explore from 2400m depth plane. It moved upward step by step. The results presented in Fig. 10 and TABLE III demonstrate that the AI technique is useful in a real environment.

TABLE I
THE RESULTS OF SIMULATION A.

Map	Covering Area (m^2)	Size of Artificial Island (m^2)	Total Path Length using (7) (m)	Total Path Length using (9) (m)
(a)	12077	0	1560	1560
(b)	11444	633 (5.2%)	1636 (+4.9%)	1560 (+0.0%)
(c)	9975	2102 (17.4%)	1635 (+4.8%)	1479 (-5.2%)
(d)	8625	3452 (28.6%)	1565 (+0.3%)	1376 (-12%)
(e)	7765	4312 (35.7%)	1517 (-2.8%)	1302 (-17%)

TABLE II
THE RESULTS OF SIMULATION B.

Map	Covering Area (m^2)	Size of Artificial Island (m^2)	Total Path Length without AI (m)	Total Path Length with AI (m)
(a)	5518	0	691	691
(b)	2421	3863 (61.5%)	805	683 (-15%)
(c)	4815	3866 (44.5%)	1157	937 (-19%)
Total			2653	2311 (-13%)

During the covering process we can construct a map using the data set obtained from the AUV's sensors. Therefore, a three dimensional terrain map of the Ulleung Basin is built with the range sensor data set.

IV. DISCUSSION

The results of simulation A imply that as the AI becomes larger, the efficiency of the covering algorithm increases. Therefore, we can infer that using large AIs makes it easy to satisfy inequality (9). Of course an AI with very complex and long boundaries does not correspond to the above tendency.

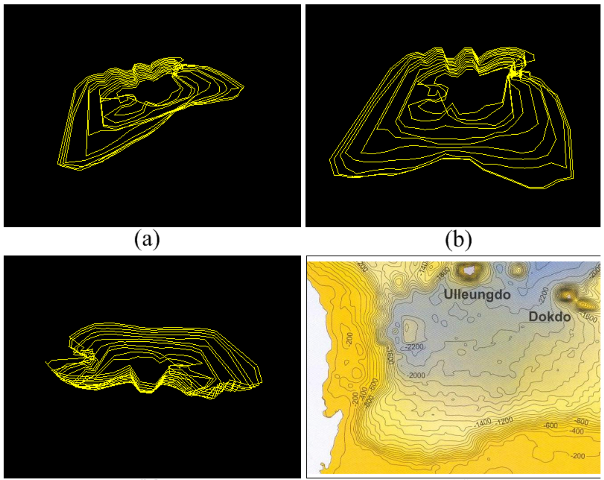
Usually as the depth increases, the total area to cover increases. Therefore, the increase in the path length in TABLE II is not a serious problem, because when inequality (9) is satisfied, the path length of the AUV using the AI technique is guaranteed to be shorter than the worst case of the path length not using the AI technique.

The result of simulation C implies that the AI technique is valid when applied in a real situation. In some planes, the AI is not generated, because the covering procedure without the AI provides a shorter path length. As the covering area becomes larger, the AI technique becomes more useful.

Through simulation A to C, a terrain map is built from the results of the covering algorithm using the proposed technique.

V. CONCLUSION

In this paper, the system architecture of the proposed



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

Fig. 10. Covering simulations of the Ulleung Basin: (a), (b), (c) three dimensional map results from different view, (d) two dimensional map of the Ulleung Basin [13]

TABLE III
THE RESULTS OF SIMULATION C.

Depth (m)	Total Path Length without AI (m)	Total Path Length with AI (m)	Running Time without AI (s)	Running Time with AI (s)
2400	683.49	683.49 (X)	455.66	455.66 (X)
2300	838.62	838.62 (X)	559.08	559.08 (X)
2200	931.4	831.51 (-10.72%)	620.93	554.34 (-10.72%)
2100	960.88	960.88 (X)	640.58	640.58 (X)
2000	999.51	962.13 (-3.74%)	666.34	641.42 (-3.74%)
1900	1328.57	1253.27 (-5.67%)	885.71	835.51 (-5.67%)
1800	1462.81	1206.33 (-17.53%)	975.21	804.22 (-17.53%)
1700	1517.4	1436.98 (-5.30%)	1011.6	957.99 (-5.3%)
1600	1694.87	1319.89 (-22.12%)	1129.91	879.93 (-22.12%)
1500	1793.59	1407.16 (-21.55%)	1195.73	938.11 (-21.55%)
Total	12211.14	10900.26 (-10.74%)	8140.75	7266.84 (-10.74%)

terrain covering and map building algorithm for an AUV in a three dimensional underwater environment is described. The artificial island technique is developed and a mathematical analysis of the technique is provided to improve the efficiency of the algorithm. In the simulations, the AUV is operated in various terrain environments. The simulation results confirm the validity of the AI technique simulator. Moreover, a three dimensional terrain map is constructed from these simulation results.

As a future work, a new algorithm that does not divide the three dimensional space into two dimensional planes will be developed. In this way, the AUV will be able to move in any direction. Therefore, the path of the AUV is determined freely. With this advantage, the path length and operating time of the AUV can be significantly reduced.

REFERENCES

[1] H. Choset, "Coverage of known spaces: The Boustrophedon cellular decomposition," *Journal of Autonomous Robots* vol. 9, issue 3, pp. 247-253, 2000.

[2] Y. Gabriely, and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot," *Proc. of the 2001 IEEE International Conference on Robotics & Automation*, pp, 1927-1933, Seoul, Korea, 2001.

[3] D. R. Blidberg, and J. Jalbert, "Mission and system sensors. In underwater robotic vehicles: Design and control," J. Yuh(Ed.), TSI Press, pp. 185-220, 1995.

[4] Y. Habriely, and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Computational Geometry: Theory and Applications*, vol. 24, no. 3, pp. 197-224, 2003.

[5] J. Oommen, S. Iyengar, N. Rao, and L. Kashyap, "Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case," *IEEE Journal of Robotics and Automation*, RA-3(6), pp. 672-681, 1987.

[6] N. Rao, and S. Iyenger, "Autonomous robot navigation in unknown terrains: Incidental learning and environmental exploration," *IEEE Transactions on System, Man and Cybernetics*, vol. 20-6, pp. 1443-1449, 1990.

[7] V. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6-4, pp. 462-472, 1990.

[8] S. Hert, S. Tiwari, and V. Lumelsky, "A terrain-covering algorithm for an AUV," *Journal of Autonomous Robots*, vol. 3, pp. 91-119, 1996.

[9] M. Hebert, and A. Jognson, "Sea map generation for autonomous underwater vehicle navigation," *Journal of Autonomous Robots*, vol. 3, num. 2-3, pp. 145-168, 1996.

[10] N. Fairfield, G. A. Kantor, and D. Wettergreen, "Real-time SLAM with octree evidence grids for exploration in underwater tunnels," *Journal of Field Robotics*, vol. 24, issue 1-2, pp. 3-21, 2007.

[11] S. Majumder, J. Rosenblatt, S. Scheding, and H. Durrant-whyte, "Map building and localization for underwater navigation," Control and Information Sciences, Experimental Robotics VII, vol. 271, pp. 511-520, 2000.

[12] http://www.whoi.edu/sbl/liteSite.do?litesiteid=7212&articleId=11314

[13] Y.J. Yeon, Y.B. Kim, J.S. Kim, J.H. Kim, and S. H. Lee, *Ocean atlas of Korea: East Sea*, National Oceanographic Research Institute, Korea, 2007