8-24-2017

3-D Coverage Path Planning for Underwater Terrain Mapping

zongyuan shen
zongyuan.shen@uconn.edu

3-D Coverage Path Planning for Underwater Terrain Mapping

Zongyuan Shen

B.S., Heilongjiang University of Science and Technology, 2014

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

At the

University of Connecticut

2017

APPROVAL PAGE

Masters of Science Thesis

3-D Coverage Path Planning for Underwater Terrain Mapping

Presented by

Zongyuan Shen, B.S.

Major Advisor_____
Shalabh Gupta

Associate Advisor_____
Shengli Zhou

Associate Advisor_____
Abhishek Dutta

University of Connecticut

2017

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor, Prof. Shalabh Gupta for his patience, excellent guidance, and constant encouragement through the process of my M.S. study. The door of Prof. Gupta's office was always open whenever I had a question about my research. I would also like to thank my associated advisors, Prof. Shengli Zhou and Prof. Abhishek Dutta for the useful comments and engagement.

Moreover, I received much help from my colleagues Junnan Song, Khushboo Mittal during my M.S. study. I would like to express my sincere thankfulness to them. Without their help, I was not able to finish my research works successfully. Also, I would like to thank my colleagues James Hare, James Wilson for their engagement and feedback on my master thesis defense.

Finally, I would like to bring my sincerest gratitude to my parents, for their support and love.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

Chapter 1

Introduction

The objective of this thesis is underwater terrain mapping. It has received much attention as it provides information about the underwater environment, which is obviously useful in many applications such as geophysical surveying, surveillance, and demining [1]. Furthermore, invisible underwater environments may pose risk to divers as well as traversing vehicles during search and rescue operations due to presence of underwater rocks. Therefore, the terrain information could be used for avoiding accidents via safe path planning. In general, the underwater terrain mapping is conducted by passing an Autonomous Underwater Vehicle (AUV) equipped with various sensors [2] over all points of a search area while avoiding obstacles by applying coverage path planning (CPP) algorithm. Thus, the use of coverage path planning algorithm is a key factor in the underwater terrain mapping problem.

In our previous work [3], we proposed an autonomous integrated system that applies a 2-D on-line coverage path planning algorithm [4] which accommodates complete coverage of an a priori unknown underwater environment by making an AUV autonomously navigate at a fixed depth from the surface, as illustrated in Fig. 1. Although the underwater terrain can be successfully reconstructed, the AUV is unable to scan certain inaccessible area such as the area above the operation

height and those beyond the sensor range, thus leading to incomplete reconstructed map. However, 3-D coverage path planning algorithm allows to AUV to scan the whole area.

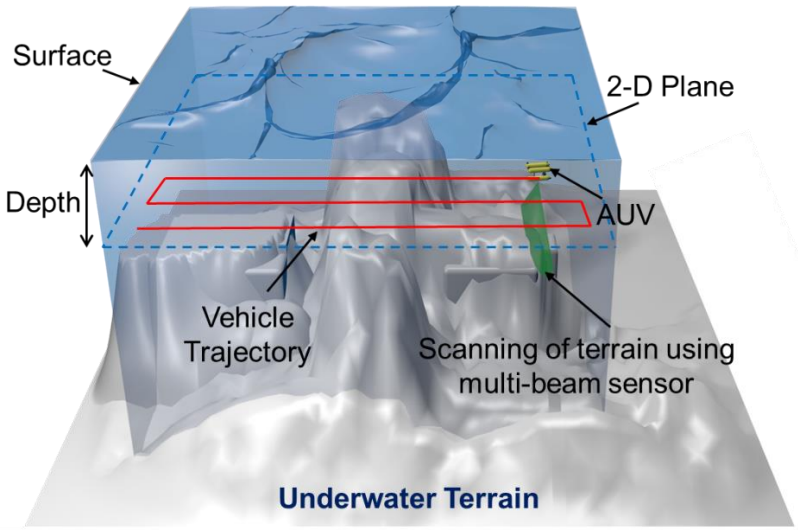

Fig. 1: 2-D coverage of underwater terrain using an AUV

In this regard, this thesis presents an on-line 3-D coverage path planning algorithm that allows the AUV to search an a priori unknown 3-D underwater environment, which is decomposed into multiple 2-D planes, at various levels. The difference between any two successive levels is kept slightly less than the maximum range of the downward-facing multi-beam sensor. At each level, by using the measurement data from the downward-facing multibeam sonar sensor as well as the AUV pose, a 2-D occupancy grid map for level below is created, which highlights the probability that each grid cell is occupied by obstacles, and the cells with the probability higher than a predefined threshold are marked as obstacle on a 2-D obstacle map. The AUV then utilizes this 2-D obstacle map to

navigate safely in the free space at that level; and consecutively generate the map for level below and so on. In this process, a multi-level coverage tree [5] [6] is incrementally built on-line, where the nodes of the coverage tree act as safe sub-areas for AUV navigation. This tree is used by the AUV for planning the coverage path of the 3-D space. Finally, the terrain data collected by the AUV is used for reconstruction of the seabed.

The proposed algorithm has been tested and validated on a high fidelity underwater simulator UWSim [7]. The main features of the proposed approach are highlighted as follows:

1) 3-D reconstruction of underwater terrain using terrain data collected from downward-facing multi-beam sonar sensor.
2) Autonomous coverage of the underwater terrain using multi-level coverage trees generated on-line.
3) Safe path planning to ensure vehicle safety by restricting the vehicle navigation to safe areas far from obstacles.

The rest of the thesis is organized as follows: Chapter 2 formulates the coverage problem for underwater terrain mapping. Review of several related works on 3-D coverage path planning is presented in the Chapter 3. Chapter 4 describes the proposed solution methodology in detail. The simulation results are presented in Chapter 5. Finally, Chapter 6 provides the concluding remarks and future work.

Chapter 2

Problem Statement

This thesis addresses the problem of reconstructing the 3-D map of underwater terrain using an AUV. In order to reconstruct this map, the AUV is made to navigate on multiple 2-D planes at different depths while using its on-board sensing systems to avoid obstacles, construct the coverage tree dynamically and collect necessary terrain data. This collected data is used for offline reconstruction of the 3-D underwater surface using alpha shapes algorithm [8]. Accurate reconstruction of the map requires complete coverage of the area above the seabed by the AUV while avoiding the obstacles.

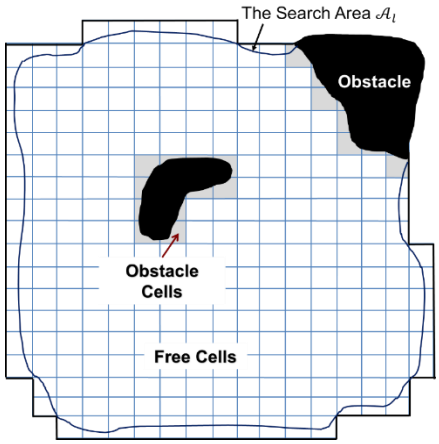

Fig. 2: Tiling of the search area $A_l \subset A$

The search space $A \in \mathbb{R}^3$ is bounded either by a hard boundary (e.g., seamount) or by a soft boundary (e.g., sub-space of a large field). An unknown number of obstacles at unknown locations are assumed to be populated inside A. This search space is comprised of multiple 2-D planes $A_l \in \mathbb{R}^2,\ l = 1,\cdots,K$, where

l refers to the navigation level and K is the maximum number of levels. A shown in Fig. 2, each area A_l is discretized into a tiling A_l^T using square cells, which are represented as τ_l^γ, where γ indicates the cell index.

Further, each cell is encoded with a state $S(\tau_l^\gamma) \in \{F, O\}$, where $F \equiv \text{Free}$ and $O \equiv \text{Obstacle}$. In order to ensure vehicle safety, the obstacle-free cells that are lying in the close vicinity of obstacles are also updated to sate O. Specifically, this is determined using the density of obstacles present in the local neighborhood of free cells. The free cells with obstacle density higher than a predefined threshold are regarded as unsafe and their states are changed from F to O. Complete coverage of A requires coverage of all τ_l^γ in the F state while avoiding the obstacles or the cells belonging to the O state. Thus, the total coverage area A^C is defined as:

$$A^C = \left\{ \cup_{l=1}^{K} \cup_{\gamma=1}^{|A_l^T|} \tau_l^\gamma : S(\tau_l^\gamma) = F \right\} \quad (1)$$

This thesis proposes autonomous navigation of the AUV to compute its trajectory *in situ* for complete coverage of A^C. It is achieved by constructing the multi-level coverage tree and traversing the entire tree using tree traversal strategies. Coverage of each node of sub-area of the tree is done using an adaptive coverage path planning algorithm [4] which guarantees complete coverage of the sub-area. The multi-beam sonar sensor data collected during the coverage of A^C is used for the reconstruction of terrain map.

Chapter 3

Literature Review

Sadat et al. [6] used a coverage tree structure for coverage path planning by using UAVs. In their work, the interesting sub-areas, each of which is treated as a node of the coverage tree, are non-uniformly distributed in the target region. For complete traversal of the coverage tree, three strategies are applied, namely the Breadth-First (BF) strategy, Depth-First (DF) strategy, and Shortcut Heuristic (SH) strategy. In their later work [9], they improved their algorithm by using Hilbert space filling curve to shorten the total length coverage path. However, the aerial coverage proposed in their works was performed in obstacle-free environment in contrast to the underwater environment containing obstacles.

Galceran et al. [10] presented an off-line approach for coverage of the 3-D underwater structures. First, they classified the terrain into high-slope areas and planar areas using a prior bathymetric map of the target area. Then, a slicing algorithm for coverage of high-slope areas is applied, while the planar areas are covered by using the AUV, following the lawnmower path at constant altitude from the seabed, where the high-slope areas are treated as obstacles. Therefore, the union of both coverage paths consist the complete coverage of the whole target area. However, the availability of prior information is difficult, particularly for unknown or partially known underwater environments.

Cheng et al. [11] proposed an off-line method for 3-D coverage of urban structures, which are simplified by abstract models, e.g., the hemispheres and cylinders in contrast to underwater complex structures. Then, these simpler structures are covered by using an UAV, following predefined time-optimal paths, which tend to perform poorly in unknown or partially unknown environments even though they may perform well under well-known environments. The approach is demonstrated in the hardware-in-the-loop simulation using a fixed-wing aircraft.

Lee et al. [12] proposed an on-line algorithm for coverage of 3-D underwater space, which is composed of multiple 2-D planes, at various depths from the ocean surface. At the beginning, the AUV starts to explore at the lowest plane. After that plane is completely covered, the artificial obstacle is added to the upper plane so as that shorten the coverage path length. The procedure is repeated until the whole environment is completely covered. By using this algorithm, the areas which have been scanned at the lower plane can be avoid scanning again at the higher plane. Moreover, the volume of water at a certain distance from the seabed is discarded and a more efficient exploration of the underwater terrain is achieved. However, this algorithm is unable to handle the case that the interesting areas are distributed non-uniformly on the underwater seabed, which means the fully interesting area is separated into multiple sub-areas by the obstacles.

In comparison, this thesis addresses the problem of terrain mapping by

adopting a novel 3-D on-line coverage path planning algorithm in an a priori unknown underwater environment containing an unknown number of obstacles at unknown locations.

Chapter 4

Solution Methodology

4.1 Overview

In our previous work [3], we proposed an integrated control architecture that controls the AUV to scan the terrain beneath it using a multi-beam sensor. The AUV is autonomously navigated on a surface located at a fixed height above the terrain, and the quality of obtained terrain information depends on the completeness of coverage of the navigation surface. However, since the multi-beam sensor is mounted to be facing downwards for terrain scanning, it will not be able to collect information from areas such as the side surfaces of underwater hills and any terrain that is above the navigation surface. In this regard, the AUV should operate at different heights to collect corresponding terrain information, and then merge them to reconstruct the whole terrain surface. However, retrieving the AUV and resetting the system for each operation height can significantly increase total mission time. Therefore, this thesis presents an advanced control system that allows the AUV to autonomously navigate at different heights to completely scan the underwater terrain for seabed reconstruction in a single run.

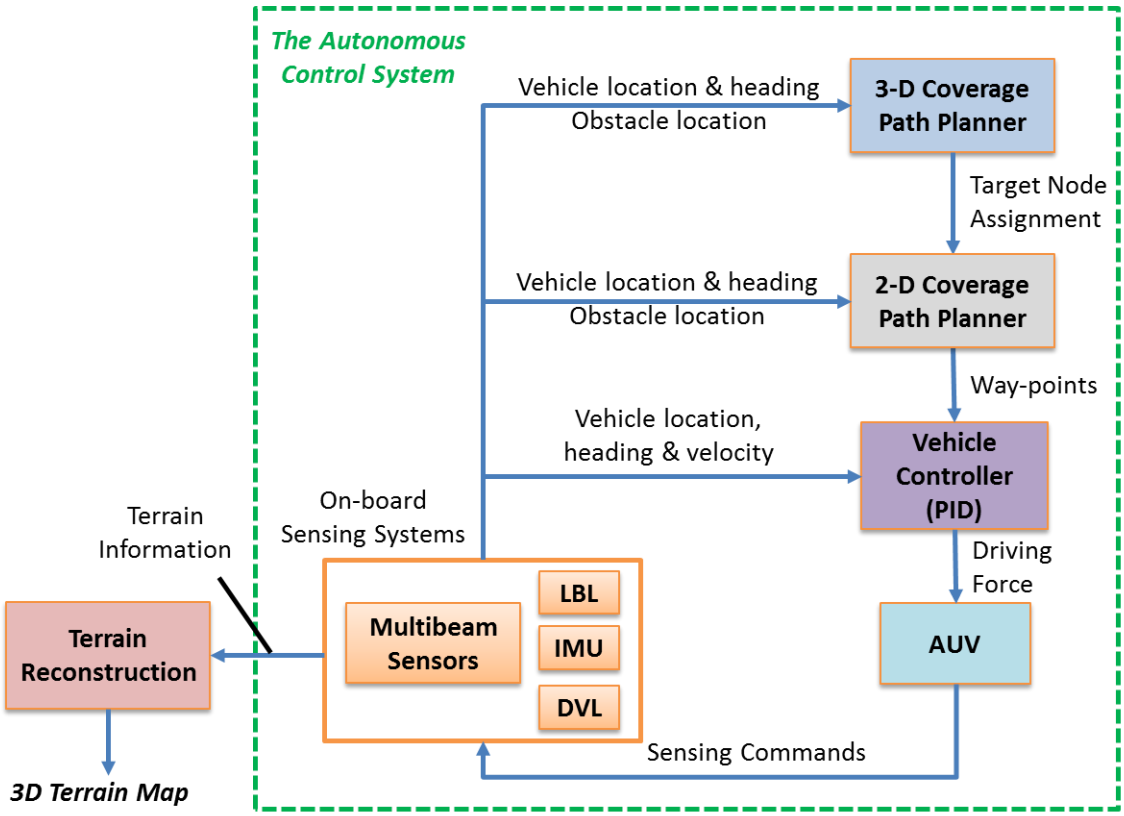

Fig. 3: Control architecture for underwater terrain mapping

Fig. 3 shows the proposed control architecture for 3-D terrain information collection and surface reconstruction. It comprises of the following subsystems:

1) The 3-D coverage path planner, which relies on the concept of multi-level coverage tree to track the search progress and compute the next unsearched targets for the AUV.
2) The 2-D coverage path planner, which receives and assignment of next area for coverage from the 3-D coverage path planner, and generates online coverage paths such that the target area is completely covered by

the AUV. In particular, this thesis adopts the multi-resolution navigation algorithm [4] for adaptive online coverage path planning.

3) The on-board sensing systems, include the multi-beam sonar sensors for obstacle detection and collection of terrain data; the long baseline acoustic system for AUV localization in GPS-denied underwater environments; and the Inertia Measurement Unit (IMU) and the Doppler Velocity Log (DVL) to measure AUV's heading and velocity, respectively.
4) The vehicle controller, i.e., a PID controller which ensures that the vehicle can reach the way-points in a safe and efficient manner.
5) Terrain reconstruction, which conduct the 3-D surface reconstruction of underwater terrain offline by using the terrain data collected online by downward-facing multi-beam sonar sensor.

The proposed autonomous control system can navigate the AUV through unknown underwater environments and collect sufficient terrain information for reconstruction of terrain map.

4.2 Vehicle Controller

The proposed methodology uses four different PID controllers as local vehicle controllers to drive the AUV to the way-points provided by the 2-D coverage path planner at stable manner. These PID controllers are as follows:

1) A depth-maintaining PID to make the AUV navigate at a fixed depth from the surface.
2) An angler PID to make the AUV turn slowly until it is facing the way-point
3) A 2-D motion PID to make the AUV move towards the way-point in a straight line.
4) A velocity PID to make the AUV stop with a very small velocity when it is close enough to the way-point.

Sensor feedback from LBL system, DVL, and IMU sensors are used for the operation of these PID controllers.

4.3 Underwater Localization

The long baseline (LBL) acoustic system [13] for underwater localization of the AUV as shown in Fig. 4. The system consists of at least four sea-floor mounted acoustic transmitters for correctly locating the AUV position. As demonstrated from

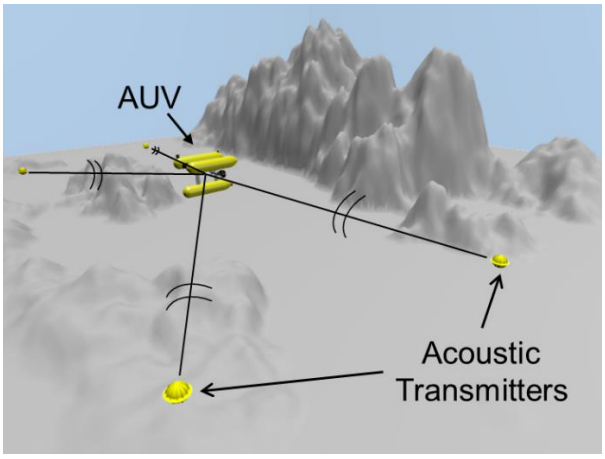

Fig. 4: The long baseline acoustic system

a wide range of applications, LBL acoustic system provides high localization accuracy and position stability which is independent of water depth. The LBL acoustic system is composed of two steps:

1) An acoustic signal is sent from each acoustic transmitter to the AUV at regular interval. The time of arrival of these signals is computed and the standard speed of acoustic signal in the water is used to convert these times to distances of the transmitters from AUV.
2) Since the position of transmitters is fixed, AUV's position can be calculated by applying the triangulation principle over these distances. When using more than four transmitters, the Least Squares method is used to get the measurement of AUV position. Finally, Kalman filter is performed on the LS measurement for a better estimation of AUV position.

4.4 2-D Coverage Path Planner

The AUV operates at a certain depth from the surface while scanning the target area provided by the 3-D coverage path planner. This requires the AUV to cover the entire planer surface it operates at. However, due to the lack of *a priori* knowledge of the underwater environment, the coverage path must adjust *in situ* when new environmental information is available.

This chapter presents the 2-D coverage path planner based on the multi-

resolution navigation controller algorithm [4]. Its key features include the guaranteed completeness of coverage, low computation burden and prevention of local minima. Specifically, this algorithm relies on the concepts of multi-resolution partitioning of the search area, local navigation and global navigation.

4.4.1 Multi-resolution Partitioning of the Search Area

The partitioning of the search area A_l aims to discretize the search area for decision-making. Let $P \triangleq \{P_\xi : \xi = 1,2,\cdots,|P|\}$ be a partition of A_l, where $\forall i,j \in \{1,\cdots,|P|\}$, it has $P_i \cap P_j = \emptyset$. Here $|P|$ refers to the cardinality of P and each element P_ξ is called a cell. Although the navigation of the autonomous vehicle is discrete, its movement remains continuous.

In order to avoid the local minima problem, this thesis constructs the multi-resolution partitioning [4] of the search area. In this thesis, a local minimum refers to the situation where the autonomous vehicle cannot find new goals even though there are still unexplored areas in the environment.

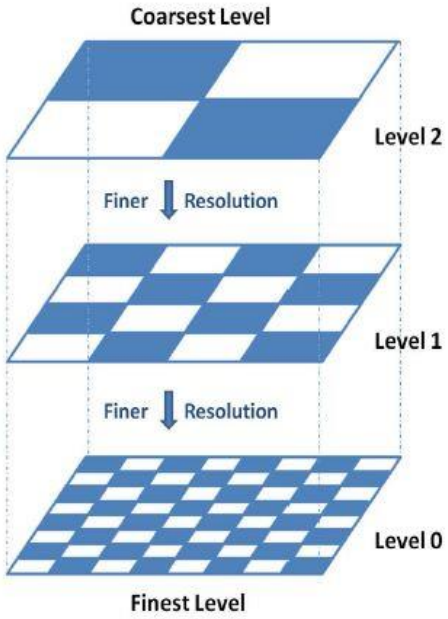

Fig. 5: An example of 3 levels of partitioning

The multi-resolution partitioning generates in total $K+1$ layers (i.e., $\text{Level } 0, \cdots, K, K \in \mathbb{Z}^+$), representing the search area with various resolutions, where $\text{Level } 0$ refers to the finest level with the highest resolution. It provides a hierarchical view of the environment map, hence enabling the autonomous vehicle to incrementally utilize the environmental knowledge for path planning and further preventing any local minimum. Fig. 5 presents an example of 3 levels of partitioning with $K = 2$. The detailed procedures to construct the multi-resolution partitioning of the search area are presented in [4].

Further, the navigation controller is designed based on the multi-resolution partitioning. It relies on the multi-resolution navigation concept, which comprises of local navigation and global navigation.

4.4.2 Local Navigation

By default, the AUV adopts the local navigation for path planning. It operates at Level 0 using the latest environment information to find movement goals. The environmental information is encoded on each cell based on its physical state. Let the set of all possible physical states be $\Delta \triangleq \{\delta_j : j = 0, \cdots, |\Delta| - 1\}$. In this thesis, Δ contains four states [14] (i.e., $|\Delta| = 4$):

1) $\delta_0 = E(Explored)$
2) $\delta_1 = U(Unexplored)$
3) $\delta_2 = O(Obstacle)$
4) $\delta_3 = B(Buffer)$

The states of all the cells are initialized with U as the search area is assumed unknown in the beginning. As the AUV explores the environment using the on-board sensing systems, the cells traversed by its trajectory are updated as *E*, while those occupied by detected obstacles are marked as *O*. Moreover, the cells lying in the close vicinity of obstacles are labeled as *B*. The purpose of constructing the buffer is to prevent the vehicle from getting too close to obstacles, so that it can safely stop or turn without any collision in uncertain underwater environments. The states of the cells evolve as the vehicle explores the environment.

Based on the encoded symbols, a potential surface is constructed by the local navigation for navigation control. Specifically, the potential energy E_{ξ}^{P}, associated to the cell ξ, with state q_{ξ}, is computed as follows:

$$E_{\xi}^{P}(t) = \Psi\left(q_{\xi}(t)\right) + \Phi\left(B_{\xi}(t), q_{\xi}(t)\right) \tag{2}$$

Where the first term on the right-hand side assigns very negative potential on obstacles and their surrounding buffet. It is defined by:

$$\Psi\left(q_{\xi}(t)\right) = \begin{cases} -\infty, & if\ q_{\xi}(t) = B\ or\ O \\ 0, & otherwise \end{cases} \tag{3}$$

The second term on the right-hand side of Eq. (1) aims to control the regular coverage motion of the AUV in an obstacle-free environment. The function Φ is defined as:

$$\Phi\left(B_{\xi}(t), q_{\xi}(t)\right) = \begin{cases} B_{\xi}(t), & if\ q_{\xi}(t) = U \\ 0, & otherwise \end{cases} \tag{4}$$

Where the time-varying potential field $B_{\xi}(t)$ is defined as:

$$B_{\xi}(t) = B + C_{\xi,\mu}(t) \tag{5}$$

Where B is an external potential field that is constructed off-line. It assigns equal potential to the cells lying in the same row, while the potentials in different columns increase gradually from top to bottom. Moreover, $C_{\xi,\mu}(t)$ computes the cost to reach a cell ξ when the AUV is currently sitting somewhere in cell μ.

$$C_{\xi,\mu}(t) \triangleq d_{tr}C_{tr} + \theta_{tu}C_{tu} \tag{6}$$

Where C_{tr} and C_{tu} are the travelling cost per unit length and turning cost per degree, respectively. d_{tr} represents the total traveling distance to the centroid

of cell ξ, and θ_{tu} is the total turning angle towards cell ξ from its current heading angle.

The autonomous vehicle incorporates the measurements of forward-facing multi-beam sonar sensor and dynamically updates the potentials for the cells lying inside its local neighborhood. In this thesis, the local neighborhood of the autonomous vehicle is defined as the 3×3 neighbor cells with the vehicle sitting at the center. The movement goal is then computed as the centroid of the cell with the highest positive potential.

However, if no unexplored cells are available in its local neighborhood, the autonomous vehicle is trapped in a local minimum. This motivates the global navigation concept to search for unexplored areas beyond its local neighborhood.

4.4.3 Global Navigation

The global navigation is used only when the autonomous vehicle gets stuck in a local minimum. In such cases, the autonomous vehicle enters a coarser level of Level 1 and its current position is converted to a corresponding position in Level 1. Every cell on coarser levels (i.e., Level $1, \cdots, K$) maintains a low-dimensional probability vector, indicating the probabilities of each state for the corresponding cells in Level 0, and it can be easily computed and stored.

The autonomous vehicle searches its neighborhood for coarser cells that have a positive unexplored probability, and the one with the highest unexplored probability is chosen. Then among the corresponding cells in Level 0, an unexplored cell is randomly selected, and its centroid is set as the next movement goal. Upon finding the new goal, the autonomous vehicle switches back to Level 0 and utilizes Bug2 Algorithm to reach it. The local navigation is then used again to proceed in the regular manner for coverage.

However, if the autonomous vehicle cannot find unexplored cells in Level 1, it further switches to Level 2 to search in an even broader area. This procedure is repeated until it finds a coarser cell with a positive unexplored probability. If no unexplored probability can be found even at Level K, it implies that the entire area has been fully searched, thereby achieving complete coverage.

4.5 3-D Coverage Path Planner

The objective of the 3-D coverage path planner is to compute the sequence of sub-areas for the AUV to cover, such that the entire search area A is covered. Since the AUV can operate at various depths, the 3-D coverage path planner relies on a multi-level coverage tree to store and track the search progress of the whole area, as well as to compute the next unscanned target sub-areas online. Next, the dynamic construction of multi-level coverage tree and the online tree traversal

strategy are presented respectively.

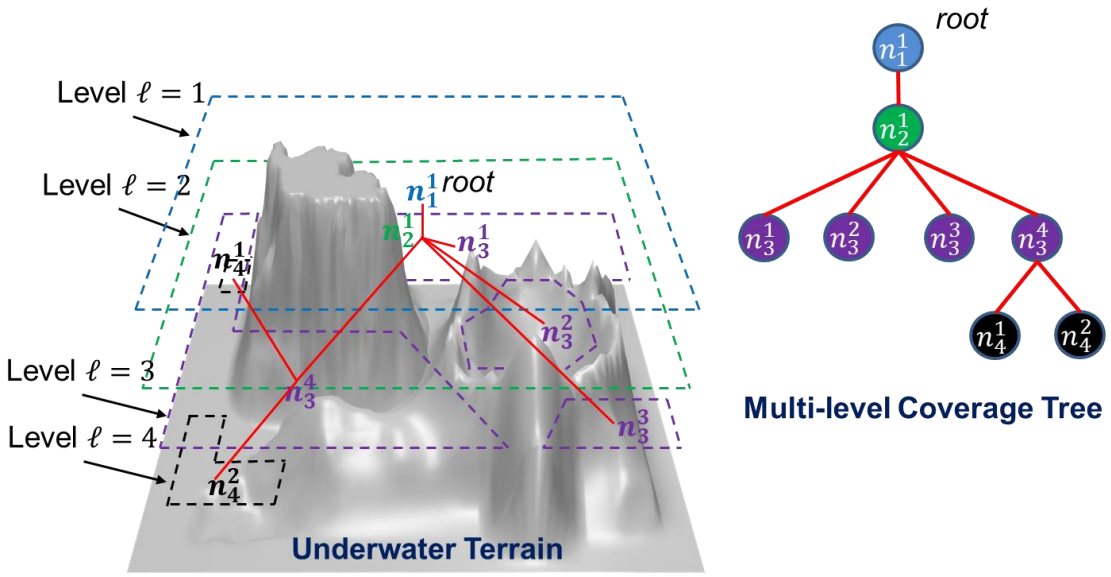

Fig. 6: An illustrative example of 4-level coverage tree

4.5.1 Multi-level Coverage Tree

Let us define the coverage tree. Each node n_l^α of the tree represents a sub-area A_l^α with index α at level $l \in \mathbb{Z}^+$. Each node has only one parent node but can have more than one child nodes, and the area it represents is disjoint from the areas represented by its sibling at the same level. The total area jointly covered by all nodes belonging to level l is $\cup_\alpha n_l^\alpha \in A_l$.

Fig. 6 shows an illustrative example of a 4-level coverage tree using BFS where the set of search areas are $\{A_l: l = 1,2,3,4\}$. At the beginning, the AUV starts at level 1, and the root node n_1^1 is the entire search area A_1. The coverage path planner enables the AUV to completely search A_1 while collecting the information

for the 3-D underwater structures that are visible to the downward-facing multi-beam sonar sensor until the level 2 and generates a probabilistic map by using occupancy grid mapping algorithm as shown in Fig. 7, where each cell holds a probability that the cell is occupied by the obstacles. Once the search area A_1 is completely covered by the AUV, the corresponding cells with the occupancy probability higher than a predefined threshold will be marked with state O on the 2-D obstacle map, which highlights the regions occupied by obstacle and the free space on that 2-D plane. To ensure vehicle safety during operation at level 2, we further compute the obstacle densities in the local neighborhood of each obstacle-free cell, and the free cells with the obstacle density higher than a predefined threshold are also marked with state O on the updated 2-D obstacle map which will be used for safe navigation at level 2. This process can exclude any potentially dangerous cell for the AUV at level 2. Then, the remaining obstacle-free cells are grouped into sub-areas based on their connectivity and each group is treated as one child node of n_1^1. For example, it can be observed in Fig. 7 that one child node of n_1^1, i.e., n_2^1 is generated and added to the multi-level coverage tree.

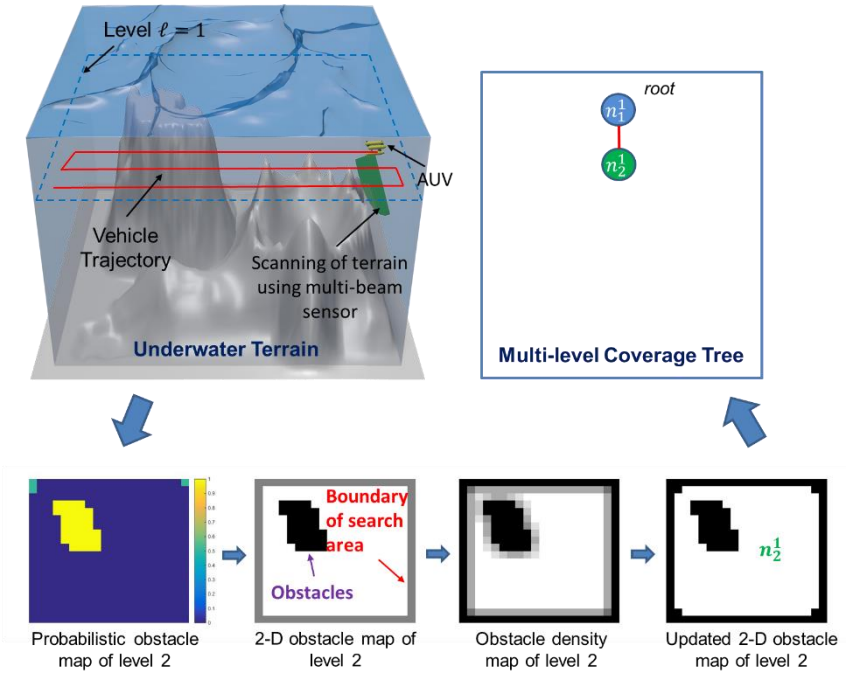

Fig. 7: An illustrative example of dynamic identification of node at level 2

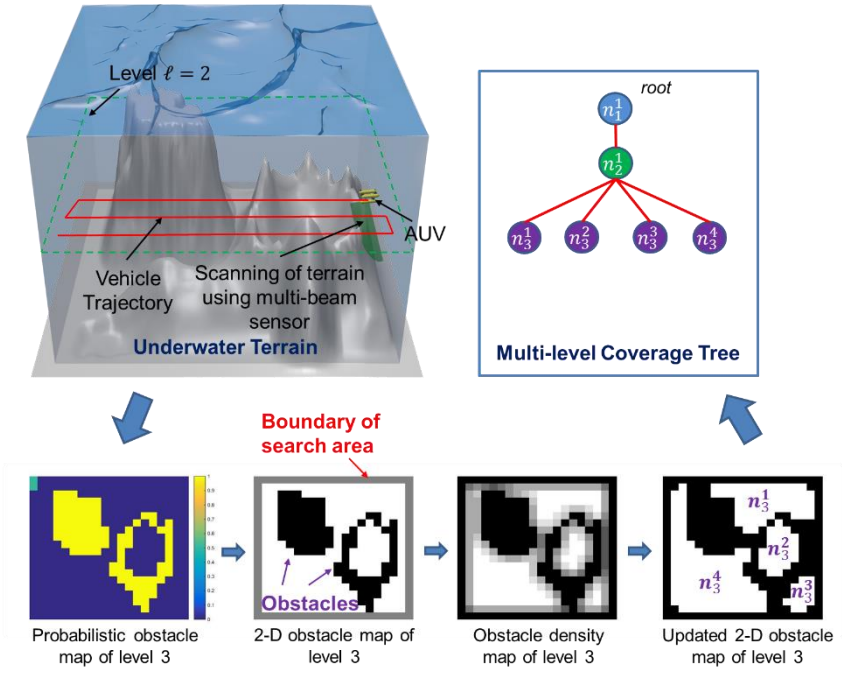

Fig. 8: An illustrative example of dynamic identification of node at level 3

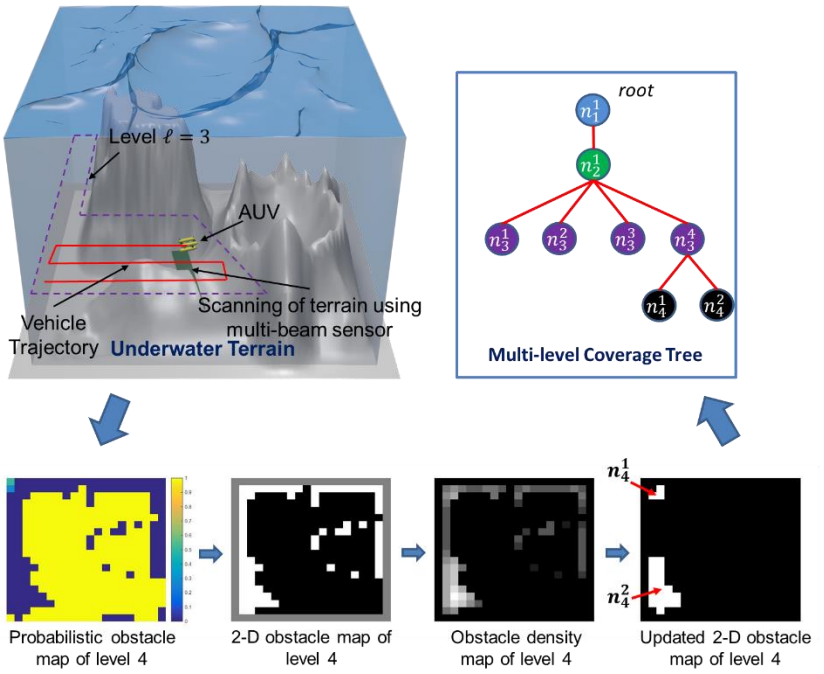

Fig. 9: An illustrative example of dynamic identification of node at level 4

During searching the sub-area A_2^1 at level 2, the AUV is able to collect the information for the 3-D terrain structures that are visible to the sensors until level 3 and thus generate a probabilistic obstacle map, where each cell holds a probability that the cell is occupied by the obstacles. Once the search area A_2 is completely covered by the AUV, the corresponding cells with the occupancy probability higher than a predefined threshold will be marked with state O on the 2-D obstacle map, which highlights the regions occupied by obstacle and the free space on that 2-D plane. To ensure vehicle safety during operation at level 3, we further compute the obstacle densities in the local neighborhood of each obstacle-free cell, and the free cells with the obstacle density higher than a predefined threshold are also marked with state O on the updated 2-D obstacle map which

will be used for safe navigation at level 3. This process can exclude any potentially dangerous cell for the AUV at level 3. Then, the remaining obstacle-free cells are grouped into sub-areas based on their connectivity and each group is treated as one child node of n_2^1. For example, it can be observed in Fig. 8 that four child nodes of n_2^1, i.e., $n_3^1, n_3^2, n_3^3, n_3^4$ are generated and added to the multi-level coverage tree.

The same procedure can be repeated when the AUV searches each of the sub-areas $A_3^1, A_3^2, A_3^3, A_3^4$, where it can further locate obstacles that would appear for Level 4 and generate corresponding child nodes as shown in Fig. 9. The coverage tree is established *in situ* as the AUV operates in the field, and the operation stops when there is no unsearched node available in the tree. Since all nodes are searched at the end, the entire area A is guaranteed to be completely covered.

4.5.2 Tree Traversal Strategies

As explained above, with the dynamic construction of the multi-level coverage tree, we need to traverse all nodes in the coverage tree online. In general, there are two standard strategies for tree traversal including the Breadth-First Search (BFS) and the Depth-First Search (DFS). Moreover, unlike the standard tree traversal strategy, we proposed an optimal tree traversal strategy which can

minimize the travel distance. These three approaches are described respectively.

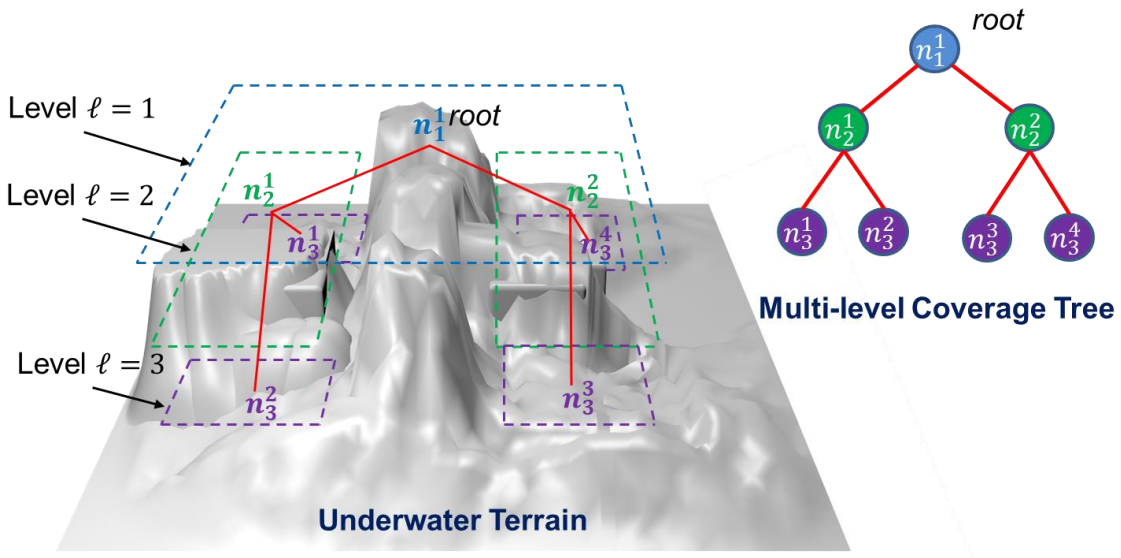

Fig. 10: An illustrative example of 3-level coverage tree

A. Breadth-First Search (BFS) Strategy

In this strategy, the AUV starts to traverse the root node n_1^1 by using the 2-D coverage path planner. During traversal of each node, the corresponding child nodes are discovered by using the on-board sensing systems and added to the multi-level coverage tree. The AUV descends to the next level and visits all the nodes at that level after visiting all nodes at previous level. In this progress, the BFS always gives the priorities to the nodes at the same level. The BFS strategy will generate the traversal path of:

$$n_1^1 \rightarrow n_2^1 \rightarrow n_2^2 \rightarrow n_3^1 \rightarrow n_3^2 \rightarrow n_3^3 \rightarrow n_3^4$$

B. Depth-First Search (DFS) Strategy

In this strategy, the AUV starts to traverse the root node n_1^1 by using the 2-D

coverage path planner. During traversal of each node, the corresponding child nodes are discovered by using the on-board sensing systems and added to the multi-level coverage tree. After visiting each node, the AUV descends to the next level to visit the child node. In this progress, the AUV explores as deep as possible along each branch before backtracking. By using the DFS strategy, the traversal path becomes:

$$n_1^1 \rightarrow n_2^1 \rightarrow n_3^1 \rightarrow n_3^2 \rightarrow n_2^2 \rightarrow n_3^3 \rightarrow n_3^4$$

C. Optimal Tree Traversal Strategy

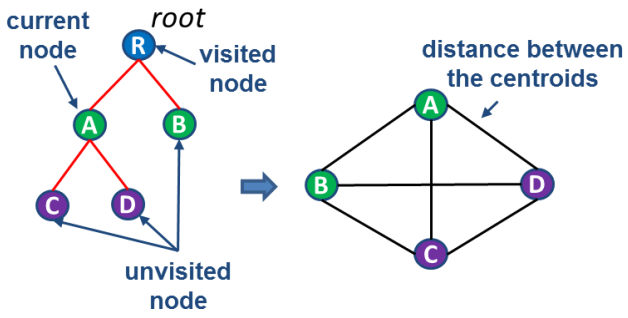

Fig. 11: An illustrative example of 3-level coverage tree and the corresponding graph

In this thesis, we proposed an optimal tree traversal strategy which can minimize the travel distance. As shown in Fig. 11, the root node R is visited node. Node A is current node while others are unvisited nodes. The objective is to find a shortest path through the graph that starts at current node A and traverses each unvisited node exactly once. The transition cost between two nodes is equal to the distance between their centroids. This is a particular Traveling Salesman Problem (TSP), and to solve this problem, we need to transform into the standard TSP first

and then perform the TSP solver to get a solution.

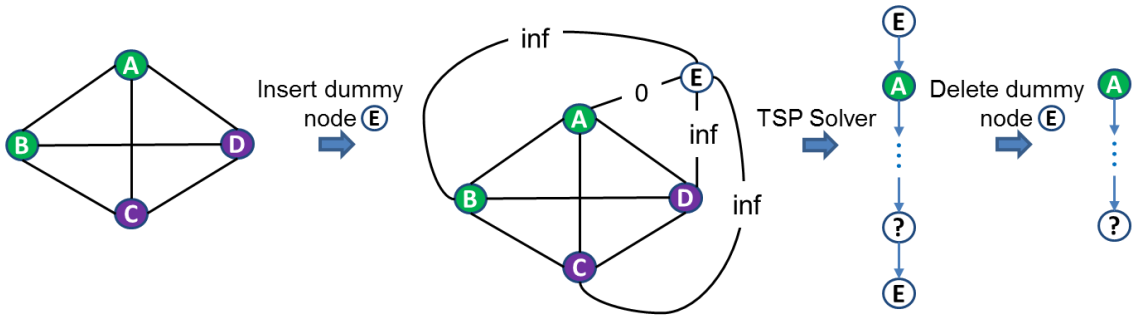

Fig. 12: An illustrative example of transformation into standard TSP

As shown in Fig. 12, a dummy node E is inserted which has a distance of zero the current node A, which we want to specify as start node in the path, while the distance between E and all other unvisited nodes is equal to infinity. The position of dummy node E in the final tour acts as the cutting point for the path [15]. Since the distance between E and A is equal to zero, the current node A will definitely list after E in the solution by applying 2-Opt heuristic. After deleting E, a path with specific start node A is achieved.

4.6 Terrain Reconstruction

In the proposed formulation, the AUV is equipped with a downward-facing multibeam sonar sensor which continuously collects the terrain data points during the vehicle navigation. Once the whole coverage tree is traversed by the AUV, the collected terrain data points are used for the reconstruction of 3-D underwater

terrain offline by sequentially applying the following approaches:

1) Outlier removal: The *k*-nearest neighbors (KNN) algorithm is used to detect the distance-based outlier [16], where the points are sorted based on their average distance to the k-nearest neighbors, and the top $n\%$ ones are regarded as outliers and then they are removed from the point set. In this thesis, we use $k = 24$ and $n = 1$.
2) Point set simplification: Once outliers are removed, a grid-based simplification method is then used to further simplify the point set, which can reduce both the storage space and calculation costs while retaining the object features. The collected point set is bounded by a rectangular bounding box which is filled with smaller 3-D cells of size $4m \times 4m$. The points located in the same cell are clustered and one point is randomly selected to be retained while the rest points are discarded. In this thesis, we processed our point set using the Computational Geometry Algorithms Library (CGAL) [17].
3) 3-D surface reconstruction: The clean data points are used to reconstruct a 3-D terrain map by using the alpha shapes algorithm [8]. This algorithm serves as a geometric tool for reconstructing the surface of an unorganized set of points. Conceptually, an alpha-shape can be regarded as a generalization of the convex hull of a point set, where the desired level of

details is controlled by a non-negative real parameter alpha as shown in Fig. 13. The MeshLab software can be used for generating the alpha-shape reconstruction.

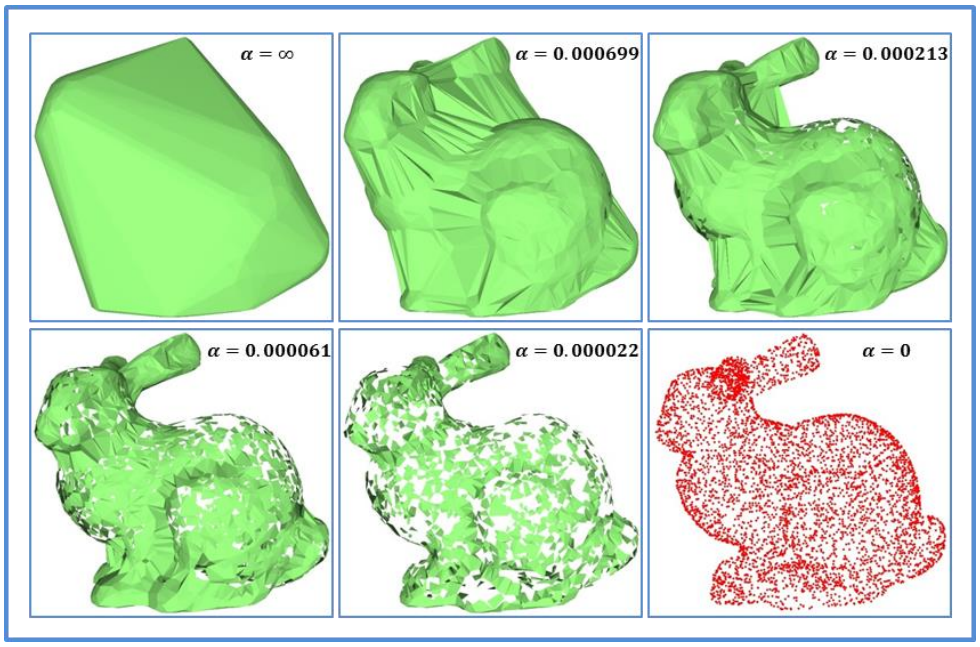

Fig. 13: An example of alpha-shape reconstructions with various alpha values

Chapter 5

Results

5.1 Simulator and Scenario Setup

The proposed method is validated on a high-fidelity simulator named UWSim [7], which is a software tool for visualization and simulation of underwater robotic tasks. The underwater virtual scenario on UWSim can be configured by using third-party modeling software such as Blender, 3-D Studio Max and so on. Controllable underwater vehicles, surface vessels and robotic manipulators, as well as simulated sensors, can be added to the scenario and interface with external control algorithm through the Robot Operating System (ROS). The snapshots of 3-D scene in the UWSim simulator are shown in Fig. 14.

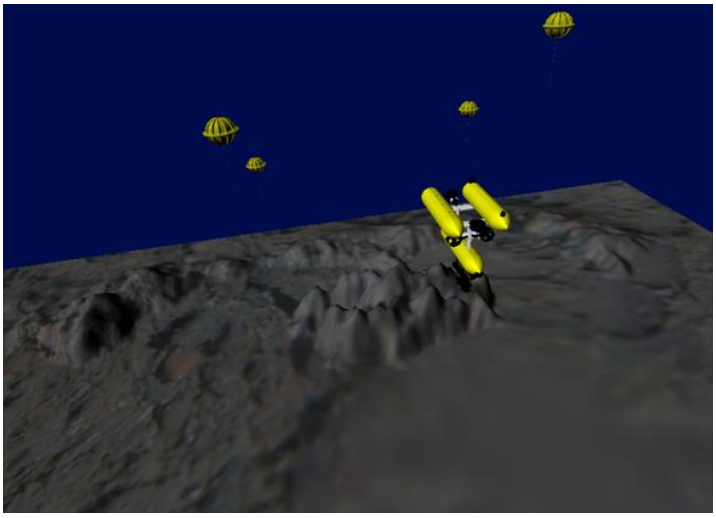

Fig. 14: A snapshot of underwater scenario in UWSim

Fig. 15 and Fig. 16 show the underwater scenarios chosen for the validation of terrain mapping algorithm. The search area is enclosed with the blue box, and

has a size of $1.8km \times 1.8km \times 1km$. For the purpose of navigation, each area A_l is partitioned into a tiling consisting of 18×18 squared cells, with each cell of size $0.1km \times 0.1km$.

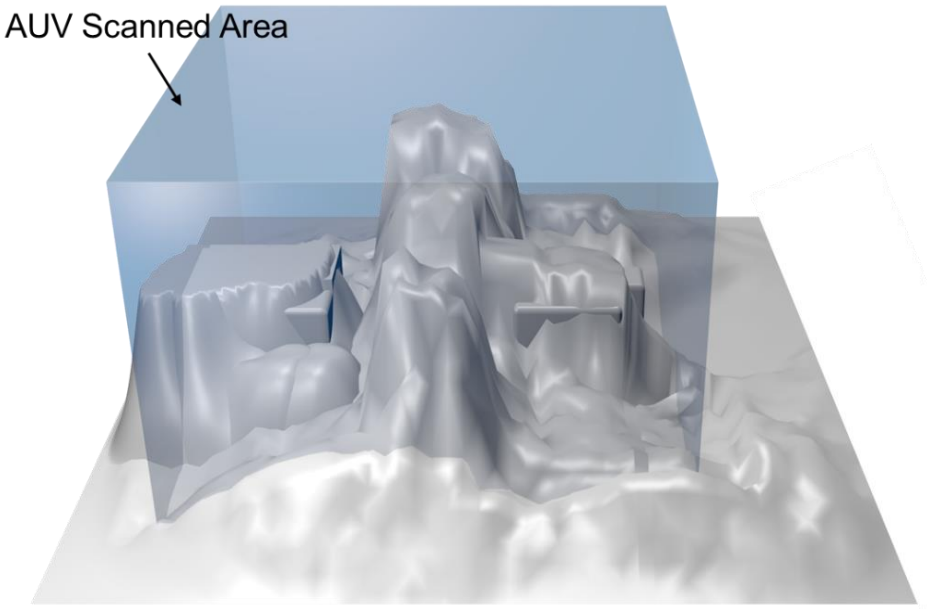

Fig. 15: Underwater scenario-1 for validation of proposed algorithm

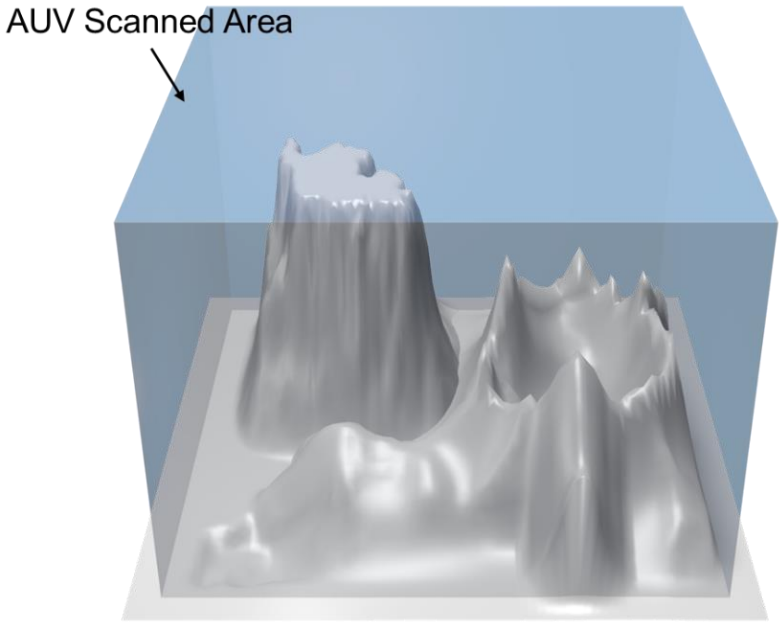

Fig. 16: Underwater scenario-2 for validation of proposed algorithm

5.2 Simulation Results on the Scenario-1

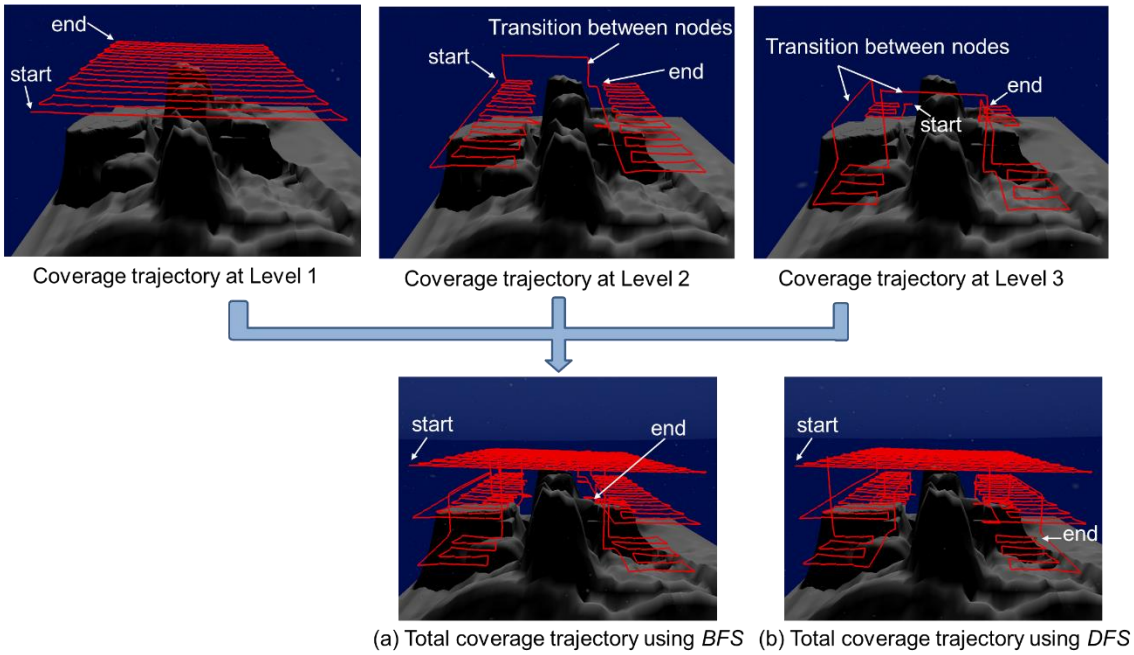

Fig. 17: Multi-level coverage trajectory using BFS and DFS

Fig. 17 shows the vehicle trajectory. The AUV is designed to operate on a total number of $K = 3$ levels, thus $A = \{A_l, l = 1,2,3\}$. For the purpose of navigation, each area A_l is partitioned into a tiling consisting of 18×18 squared cells, with each cell of size $0.1km \times 0.1km$. The operation depths $d(l)$ for each level are: $h(1) = 1.56km, h(2) = 1.86km, h(3) = 2.16km$ The depth difference between adjacent levels should not exceed the maximum range of the downward-facing multi-beam sonar sensor. Moreover, it must ensure that the whole terrain information available for Level l should be obtained by the AUV while it is operating at Level $l - 1$.

At the beginning, the AUV starts at the root and begins to search A_1.

Thereafter, it utilizes the 2-D coverage path planner to search and cover all three areas A_l, $l = 1,2,3$, as explained in chapter 4. Bother BFS and DFS traversal strategies are utilized to visit all nodes in the coverage tree online. The corresponding AUV trajectories are presented in Fig. 10 where the three figures in the left column indicate the separate coverage trajectories using BFS at each level. They compose the whole coverage trajectory together. The bottom figure in the right column in Fig. 16 shows the vehicle trajectory using the DFS strategy. The total trajectory lengths using each strategy are presented in Table 1. It can be observed that in this scenario, the DFS strategy achieves a shorter coverage path.

Table 1: Comparison of trajectory length

Tree Traversal Strategy	Total Trajectory Length
DFS	3.695*km*
BFS	3.864*km*

During the AUV navigation, the downward-facing multi-beam sonar sensor equipped on the AUV, collects the terrain data points continuously. Once the whole coverage tree is traversed by the AUV, the collected terrain data is processed by using *k*-nearest neighbors (KNN) algorithm and grid-based simplification method based on the Computational Geometry Algorithm Library (CGAL).

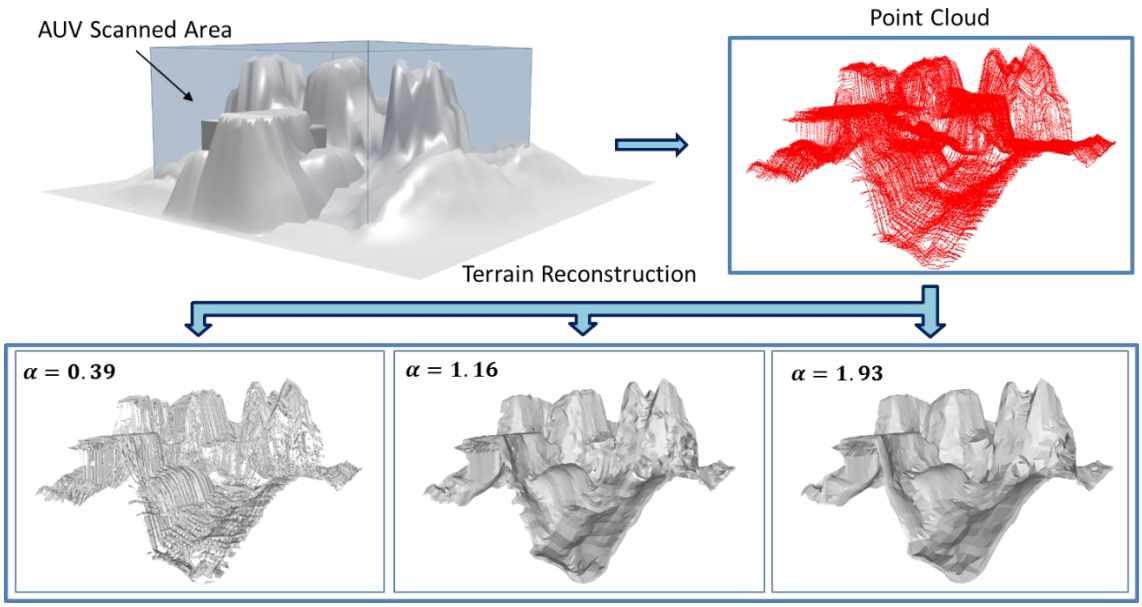

Fig. 18: Terrain reconstruction based on the point cloud obtained using multi-level BFS coverage

Then the alpha shapes algorithm [8] is applied on the clean point set for 3-D terrain map reconstruction using the MeshLab software. The final reconstructed surface with variant levels of details obtained after adopting these algorithms is shown in Fig. 18. The left figure on the upper row shows the selected scan area which is enclosed by the blue boundaries and has dimensions $1.8km \times 1.8km \times 1km$. The red points in the reconstructed figure refer to the clean data points and the grey surface is the reconstructed surface. The reconstructed figure preserves the underlying 3-D structure of the terrain, thus validating the proposed methodology.

5.3 Simulation Results on the Scenario-2

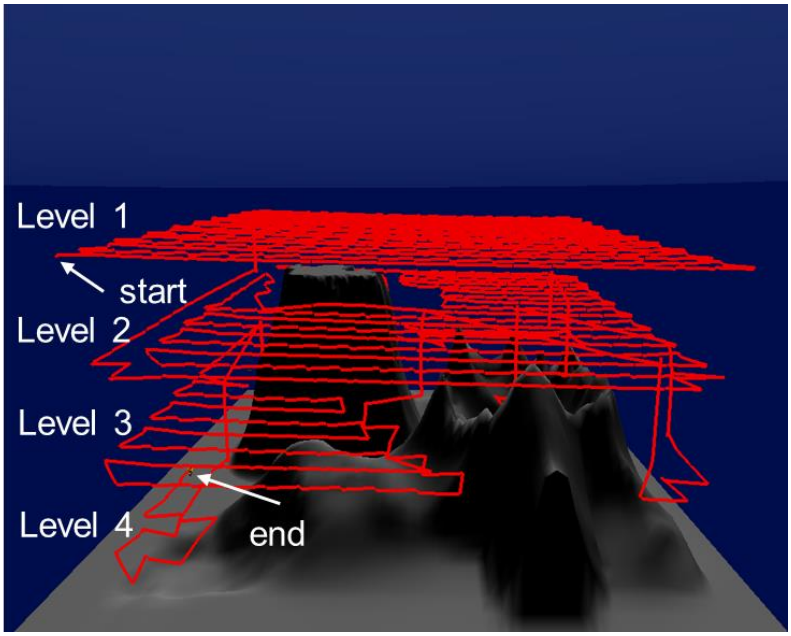

Fig. 19: Multi-level coverage trajectory using optimal tree traversal strategy

Fig. 19 shows the vehicle trajectory. The AUV is designed to operate on a total number of $K = 4$ levels, thus $A = \{A_l, l = 1,2,3,4\}$. For the purpose of navigation, each area A_l is partitioned into a tiling consisting of 18×18 squared cells, with each cell of size $0.1km \times 0.1km$. The operation depths $d(l)$ for each level are: $h(1) = 1.56km, h(2) = 1.86km, h(3) = 2.16km, h(4) = 2.46km$. And the distance between adjacent levels should not exceed the maximum range of the downward-facing multi-beam sonar sensor. Moreover, it must ensure that the whole terrain information available for Level l should be obtained by the AUV while it is operating at Level $l - 1$.

At the beginning, the AUV starts at the root and begins to search A_1.

Thereafter, it utilizes the 2-D coverage path planner to search and cover all four areas A_l, $l = 1,2,3,4$, as explained in chapter 4. The optimal tree traversal strategy is utilized to visit all nodes in the coverage tree online. The corresponding AUV trajectories are presented in Fig. 19.

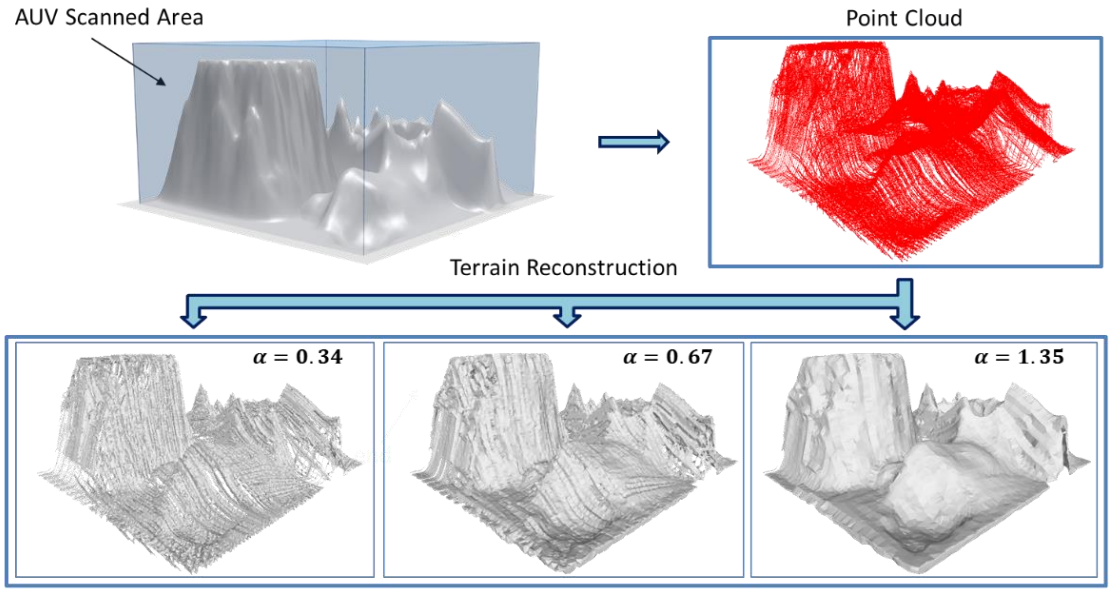

Fig. 20: Terrain reconstruction based on the point cloud obtained using multi-level optimal coverage

During the AUV navigation, the downward-facing multi-beam sonar sensor equipped on the AUV, collects the terrain data points continuously. Once the whole coverage tree is traversed by the AUV, the collected terrain data is processed by using *k*-nearest neighbors (KNN) algorithm and grid-based simplification method based on the Computational Geometry Algorithm Library (CGAL). Then the alpha shapes algorithm [8] is applied on the clean point set for 3-D terrain map reconstruction using the MeshLab software. The final reconstructed surface with variant levels of details obtained after adopting these algorithms is shown in Fig.

20. The left figure on the upper row shows the selected scan area which is enclosed by the blue boundaries and has dimensions $1.8km \times 1.8km \times 1km$. The red points in the reconstructed figure refer to the clean data points and the grey surface is the reconstructed surface. The reconstructed figure preserves the underlying 3-D structure of the terrain, thus validating the proposed methodology.

Chapter 6

Conclusions and Future Work

In this thesis, a novel approach for 3-D reconstruction of seabed terrain using multi-level coverage trees for an autonomous system is proposed. The 3-D search space comprises of various 2-D navigation levels and is modeled as a coverage tree. The performance of both standard tree traversal strategy and optimal tree traversal strategy for this coverage tree are presented. These strategies ensure that all the nodes of the tree are visited by the AUV. Moreover, each node of the tree is scanned using a complete coverage algorithm, thus leading to complete coverage of the entire search space. Vehicle safety during the AUV navigation is ensured by identifying the free spaces with high obstacle density in their neighborhood and adding them to the obstacle map. During the coverage, terrain data is constantly collected by the down-facing multi-beam sonar sensor mounted on the AUV which is later used offline for 3-D reconstruction of the seabed. The proposed method is validated to accurately reconstruct the terrain maps using a high-fidelity UWSim simulator. For future work, the sequence of traversal of the nodes of the tree can be optimized to achieve the coverage in shortest time. Other scopes for improvement include online terrain reconstruction and extension of the algorithm to incorporate the use of multiple AUVs.

References

[1] K. Mukherjee, S. Gupta, A. Ray, and S. Phoha, “Symbolic analysis of sonar data for underwater target detection,” IEEE Journal of Oceanic Engineering, vol. 36, no. 2, pp. 219-230, 2011.

[2] C. Emanuel, D. Zaidi, S. Gupta, and S. Zhou, “Integrated command, control and communication for autonomous underwater vehicles,” in OCEANS 2015, MTS/IEEE Washington. IEEE, 2015, pp. 1-5.

[3] Z.Shen, J. Song, K. Mittal, and S. Gupta, “An autonomous integrated system for 3-d underwater terrain map reconstruction,” in OCEANS 2016, MTS/IEEE Monterey. IEEE, 2016, pp. 1-6.

[4] J. Song, and S. Gupta, “Slam based shape adaptive coverage control using autonomous vehicles,” in The 10th System of Systems Engineering Conference (SoSE). IEEE, 2015, pp. 268-273.

[5] Z.Shen, J. Song, K. Mittal, and S. Gupta, “Autonomous 3-d mapping and safe-path planning for underwater terrain reconstruction using multi-level coverage trees,” in OCEANS 2017, MTS/IEEE Anchorage. IEEE, 2017, pp.1-6.

[6] S. A. Sadat, J. Wawerla, and R. T. Vaughan, “Recursive non-uniform coverage of unknown terrains for uavs,” in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 1742-1747.

[7] M. Prats, J. Pérez, J. J. Fernández, and P. J. Sanz, “An open source tool for simulation and supervision of underwater intervention missions,” in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 2577-2582.

[8] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes.” ACM Transactions on Graphics (TOG), vol. 13, no. 1, pp. 43-72, 1994.

[9] S. A. Sadat, J. Wawerla, and R. Vaughan, “Fractal trajectories for online non-uniform aerial coverage,” in 2015 IEEE/RSJ International Conference on Robotics and Automation. IEEE, 2015, pp. 2971-2976.

[10] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, “Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles,” Journal of Field Robotics, vol. 32, no. 7, pp. 952-983, 2015.

[11] P. Cheng, J. Keller, and V. Kumar, “Time-optimal uav trajectory planning for 3d urban structure coverage,” in 2008 IEEE/RSJ International Conference on Robots and Systems. IEEE, 2008, pp. 2750-2757.

[12] T.-S. Lee, J.-S. Choi, J.-H. Lee, and B.-H. Lee, “3-d terrain covering and map building algorithm for an auv,” in 2009 IEEE/RSJ International Conference on Robots and Systems. IEEE, 2009, pp. 4420-4425.

[13] L. Paull, S. Saeedi, M. Seto, H. Li, “AUV navigation and localization: a review,” IEEE Journal of Oceanic Engineering, vol. 39, no. 1, pp. 131-149, 2014.

[14] S. Gupta, A. Ray, and S. Phoha, “Generalized ising model for dynamic adaption in autonomous systems,” European Physics Letters (EPL), vol. 87, p. 10009, July 2009.
[15] M. Hahsler, and K. Hornik, “Tsp-infrastructure for the traveling salesperson problem,” Journal of Statistical Software, vol. 23, no. 2, pp. 1-21, 2007.
[16] E. M. Knox and R. T. Ng, “Algorithms for mining distancebased outliers in large datasets,” in Proceedings of the International Conference on Very Large Data Bases. Citeseer, 1998, pp. 392-403.
[17] P. Alliez, L. Saboret, and N. Salman, “Point set processing,” CGAL User and Reference Manual, vol. 3, 2010.

Appendix A: Publications

Conference Papers

[1] **Z.Shen**, J. Song, K. Mittal, S. Gupta. “An autonomous integrated system for 3-D underwater terrain map reconstruction,” in OCEANS 2016 MTS/IEEE, Monterey, IEEE, 2016, pp. 1-6.

[2] **Z.Shen**, J. Song, K. Mittal, S. Gupta. “Autonomous 3-D mapping and safe-path planning for underwater terrain reconstruction using multi-level coverage trees," in OCEANS 2017 MTS/IEEE, Anchorage, IEEE, 2017, pp. 1-6.

Journal Paper

[3] **Z. Shen**, J. Song, K. Mittal, and S. Gupta. “3-D coverage path planning algorithm for underwater terrain mapping” (in production)