Three Dimensional Path Planning and Obstacle Avoidance: An Overview

Duaa A. Ramadhan
Electrical Engineering Dept.
University of Basrah
Basrah, Iraq

Abdulmuttalib T. Rashid
Electrical Engineering Dept.
University of Basrah
Basrah, Iraq

ABSTRACT
This paper presents a survey for a three dimensional path planning algorithms which produced significant attention for the last years. It is dependent on the static and dynamic obstacles when the mobile robot draw it is trajectory to the goal. Also this paper discusses the type of the three dimensional vehicles: The Unmanned Aerial Vehicle (UAV) as a flying robot and the Autonomous Underwater Vehicles (AUVs) as a swimming robot. Two types of data structure are discussed in this paper which represents the navigable area of a virtual environment: the Voxel grid and the volumetric navigation mesh. The differences among the surveyed approaches are discussed and the results are summarized.

General Terms
Path planning, mobile robot.

Keywords
Three dimensional path planning; Multi robot; Obstacle avoidance.

1. INTRODUCTION
There are several problems must solve it if want to achieve any goal by the robot. These problems are: knowing of initial position of the robot, and the problem path planning from initial to the destination position, then deal with obstacle avoidance and collision avoidance when robot move to the destination position. There are many ways of localization have been designed in order to work in local and global environments with different types of range sensors like sonar, ultrasonic sensor, IR sensor, and laser range finders [1-5]. These ways of localization are important to produce the initial position and direction of each robot in an environment. Also, there are many ways of path planning to design a multi-mobile robot. These ways can be classified into global and local methods depend on the sensor information available [6, 7]. The global method ensures that the robot reaches to the destination while the local method which depend on sensing range of the sensors do not ensure the reaching of the robot to the destination [8]. There are open-source solutions for solving problems of metric 2D path planning for indoor robots [9-10], while in outdoor robots like autonomous cars same solutions applied but the problem mostly lies in perception and the embedding of driving rules [11]. Three dimensional path planning for aerial [12-13] or underwater [14] robots leads to stochastic approaches that deals with increased dimensionality roadmaps [15] and exploring random trees [16-17]. As Stumm [18] it considers a small robot with magnetic wheels to check up steel pipes from the inside. This robot is moving only on the surface but this surface is complex 3D object that cannot be described with elevation maps accurately. They are using voting to reconstruct representation of 3D environment based on point clouds, then using A* algorithm to search about path.

2. STATIC AND DYNAMIC OBSTACLE
2.1 Static Obstacle Avoidance
Static obstacle avoidance is a way of avoiding obstacles that is static in shape and position. There are many examples of static obstacles like mountains and fences, terrain, and the walls of a building. Also static obstacles can represent as gaps or pits that robot can fall into it or can be as borders between countries that are not even physically marked. In many applications, in order to speed up real-time route and find path around a variety of different static obstacles and based on existing information about the environment, pre-calculated data structures can be formed .

2.1.1 Area System
The system is presented here is same as system in [19] for dealing with static obstacles. In particular, this system assumes that robot lives in a six axis aligned bounding box relatives to reference point. Robot has limited number of degrees freedom. First step of off-line compilation process involves formation of boundary representation of configuration space(C-space) and this boundary configuration consists of one or more triangle meshes. Constructive solid geometry (CSG) operations are executed on two manifold triangle meshes in order to calculate the configuration space. System presented in [19] uses 3D binary space partitioning algorithm for calculating configuration space but using 3D BSP algorithm with many polygons involves errors. Because of these errors, boundary representation of configuration space may formed that does not describe complete movement freedom of robot. So before using CSG operations, it must divide each polygon into convex polytope. Each convex polytope represented by two triangle mesh. The convex polytope for a polygon is formed by calculating the convex hull of all the corners of all the positioned bounding boxes. In the next step of the off-line compilation process, it determines traversable surfaces on the boundary representation of configuration space. Then determine the slope of each triangle. If slope less than 45 degrees then triangle is considered traversable. Two dimensional 2D binary space partitioning (BSP) algorithm is used to divide traversable surfaces into areas. Two dimensional BSP algorithm is consider robust than 3D BSP algorithm because only vertical planes are used to split two manifold meshes.

2.1.2 Avoiding Ledges
The robot movement is designed as an axis-aligned bounding box, so if this bounding box is hanging upon ledge that means there is only one corner of box located over ledge and this means the box will not tumble. After that it constructs vertical plane through a ledge. This plane is moved away from the ledge because if the robot stays in front of the plane, the robot

bounding volume cannot dangle upon the ledge. With this plane the areas of area system are cut up to create areas on both sides of the plane. Areas that are located behind plane is marked as ledge areas because when robot calculate routes, it will avoid these areas.

2.1.3 Reachability
Reachability is consider a way that make robot travel from one area to the next or to an adjacent area or to the area which close proximity in a straight line. Reachability is represented as a link connect between two areas and a point in space. Areas in the area system are represented by polygons. In order to define reachability, polygon edges of adjacent areas are exactly on top of each other are used for this reason to make transaction from one area to another smoothly. There are a lot of small height differences between areas, 2D BSP algorithm is used to solve this problem. 2D BSP algorithm cut up the traversable two-manifold mesh. It can test areas in close proximity for edges that overlap when falling it onto a horizontal plane, then calculate vertical distance between the overlapping parts of such edges in order to find the places where a robot can be expected to cover the height distance and navigate across the edges from one area to the next.

2.1.4 Routing
In order to calculate routes between areas, it can do that by using reachability from one area to another. There are a type of algorithms like conventional routing algorithms can consume time when are executed with areas that contain thousands number of polygons in real-time. Also all routes can pre-calculated but this way consume time and storage space. So routes are calculated by using hierarchical routing system with the same accuracy as conventional routing algorithm because the hierarchical system takes less storage space and consume less time. In order to avoid recalculated routes repeatedly the calculated routes are temporarily saved. Typically, routes are calculated and saved at the goal because goal is stay same period of time longer than the area that robot is in.

2.1.5 Path Optimization
Fig. 1 below contains arrows that represent a path retrieved from the area system around two wall sections. Path retrieved from area system is defined as a sequence of areas connected by reachability. This is not the optimal path and can be optimized by drawing direct line from start position to one of the intermediate points. Robot must move along this line without any obstacle. Polygon can used to test that line goes through free space without any obstacle or not. Robot can choose furthest away intermediate point on the path which robot can move to it in straight line. Instead of choosing intermediate points only, it can sup-sample lines between successive intermediate points then test if robot can reach or not. By this way and by optimize path continuously, robot can find smooth path and close to optimal path.

2.2 Real-Time Dynamic Obstacle Avoidance
The area system provides a robust solution for static obstacle avoidance. When the system has many information and records about dynamic obstacles like positions and dimensions, it will calculate the areas of area system for a particular configuration of such dynamic obstacles. The system is same as bug’s algorithm or wall following but there is a difference that the path around obstacle can re-calculated and optimized repeatedly. Dynamic objects that the system deals with is represented by oriented bounding boxes (OBB). All dynamic obstacles can be contained in one or more oriented bounding boxes and this does not cause any problem. Dynamic obstacle avoidance system will always find a path around dynamic obstacle but won’t find the path to make robot move over these obstacles.

3. THE NAVIGABLE SPACE
3.1 Voxel grid
An extension of 2D regular grids to 3D is usually called a voxel grid, voxel set, or structured volumetric data set [20]. Such data sets are already used in many application that use complex 3D representations of structures, e.g. the data retrieved by some medical scans. Such 3D voxel grids are also used for volume rendering. When the data is taken from scans, it will normally be done on samples taken at equally spaced points in all dimensions (x, y and z). For games and other simulations, these voxels can be used to represent complex terrain features, e.g. caves. Yet, it should be noted that there are different ways to interpret such data structures. However, we are not directly concerned with rendering, so this will not be pursued further. This means that no assumptions will be made about how the environment is exactly represented, so the methods presented here will be usable on both environments represented by voxel structures or by mesh structures. Basically, we will create new structures which are approximations of these environment representations. These will be either stored as a voxel structure or as a mesh structure. These voxel structures can be implemented by data structures like octrees as in Fig. 2 or k-d trees. These data structures organize their data in a tree-based way; so that actions like searching, adding or removing objects, can have a good average time complexity: O (logn) with n the amount of cells. The space complexity can also be greatly reduced when using intelligent data structures like for example the sparse voxel octree as described by Laine et al. [21]. This keeps track of only the surfaces of the environment and not all inaccessible voxels. An extension of the probabilistic occupancy model takes advantage of probabilistic occupancy model characteristics that can be used to build repulsive and attractive potential fields in a three-dimensional voxel grid structure. This approach allows for an enlargement of the space that is used for the planning, especially within narrow areas. His experiments show that the use of a probabilistic model as a multiresolution structure can lead to a reduction of up to 40% of the computation time in general. This strategy appears to be valid when extended into a full 3D space. The main problem of using voxel grid structures for path planning is the amount of nodes that have to be considered along a path. While the use of multiresolution grids can greatly reduce the amount of work, it usually has still too many nodes to quickly execute a path planning algorithm on them when we have multiple agents. Also, the use of probabilistic occupancy models is actually a rough estimation. This approach is quite dependent on how _ne or coarse a grid of each level of resolution is. The coordinates considered for path planning are usually only the voxel centers or corners. Coarse grid cells may lead to better time performances, but the loss of environmental information might be too big for good results. Furthermore, it is also possible that these voxels are not aligned with the actual environment.

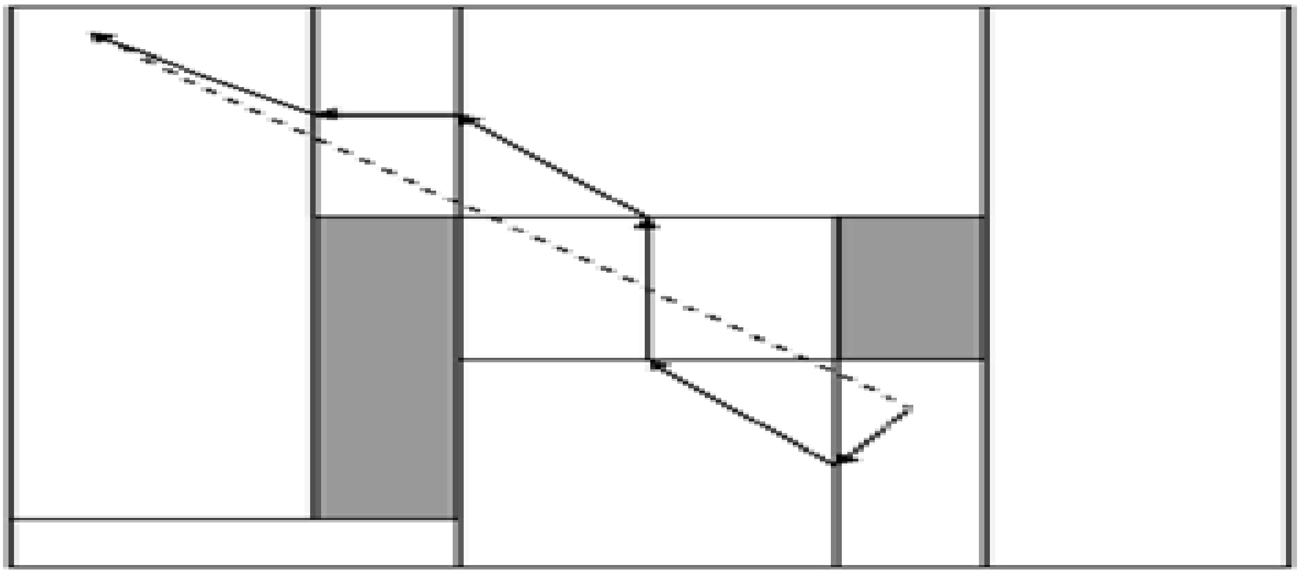

Fig 1: Path optimization around two wall sections.

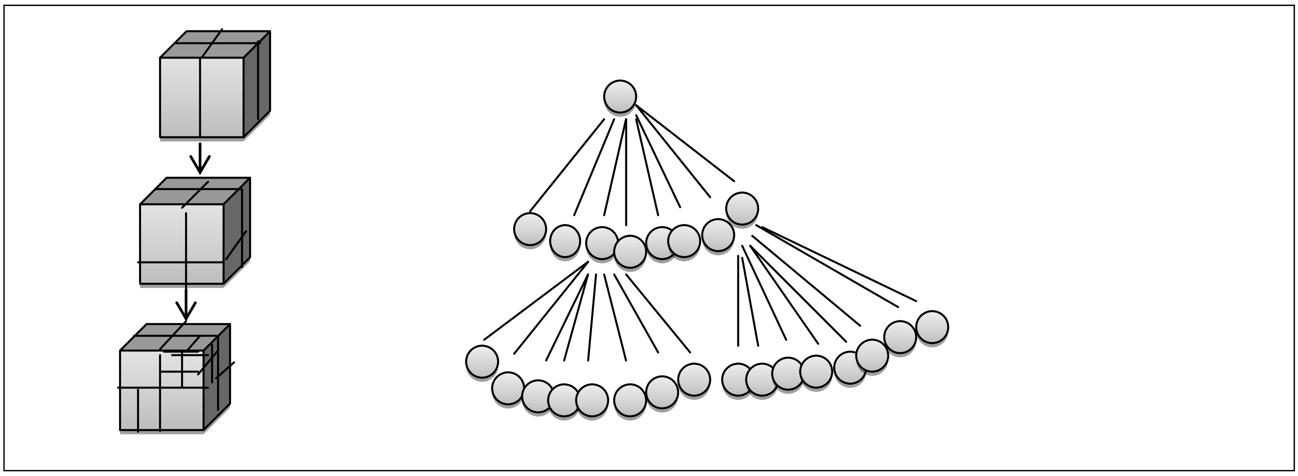

Fig 2: Octrees

3.2 Volumetric navigation mesh

3.2.1 Definition

A volumetric navigation mesh, or volumetric navigation mesh, is defined as a data structure which represents the navigable area of a virtual environment, where: the environment has to be divided in convex 3D polyhedral, also the entire mesh must be contiguous, with all adjacent polyhedral sharing a single face, also no two polyhedral should overlap on the same space. One of the ideas behind navigation meshes is to model a three-dimensional environment as a two-dimensional structure, since it is usually just used for navigation over a surface. However, his definition introduces new problems when more complex environments are used, or when full three-dimensional navigation is required. Different layers of navigation meshes could be used and connected to model multilayered environments [22]. However, this approach is mainly limited to modeling the walkable areas. We can do a better expansion for 3D for areas with volumetric obstacles. Let's call this new data structure a volumetric navigation mesh.

3.2.2 Generating the structure

There are many approaches for generating a volumetric polygonal mesh structure in 3D environments, such as the Adaptive Space Filling Volumes 3D algorithm. It works by seeding the world space with a series of unit cubes. These cubes then grow as big as possible. There is also an automatic subdividing system in it that allows conversion of these cubes into higher-order convex polygons. They show that their method is less complex and provides better results compared to some other approaches like Space-lling Volumes and Automatic Path Node Generation mesh methods. Another approach that we propose is to start with a mesh consisting of only one big convex polyhedron, normally a rectangular cuboid that contains the whole area. Then we can look at each obstacle in the environment at a time, and split up the convex polyhedron(s) it is currently in, according to the characteristics of that obstacle. This can be seen as starting with an empty environment then keep adding the obstacles, while doing local operations to update the navigation mesh.

4. D UAV TRAJECTORY PLANNING

AUVs usually operate in dynamic, cluttered, and uncertain ocean environments. In order to ensure a safe and efficient operation of vehicle in such environment, it must use real-time path planning. A path planner must react quickly with any change in environments and find a suitable path that safely moves the AUV from its initial position to its destination with minimum energy/ time cost [23]. There are many algorithms applied to AUV path planning problem like Dijkstra's algorithm, A* algorithm, Field D* algorithm, Fast Marching (FM) algorithm, and Artificial potential field. When AUVs operate in a large geographical area, path planning process is consider a large – scale optimization problem because there are computational requirements that make path planning process is slow in high dimensional search space. So if it wants to speed up planning process and reduce the memory requirement, it must use a way to translate the 3D space to 2D space but this 2D space cannot contains all the 3D information of the ocean environment. Evolutionary algorithm is consider that is efficient and effective way that used to deal with non-deterministic polynomial time (NP) hard problems [24]. Genetic algorithm (GA) [18-20] and Particle Swarm Optimization (PSO) are considered as a well-known effective optimization techniques of evolutionary algorithms that used to solve path planning problem. There is an improved version of PSO called Quantum-behaved Particle Swarm Optimization (QPSO). This version assumes that every particle swarm has quantum behavior instead of conventional position and velocity updated rules employed in PSO.

5. UNDERWATER PATH PLANNING

There are many applications underwater like oil and gas production, underwater infrastructure monitoring, coastal surveillance, ocean exploration, and weather services [21-22-23]. Autonomous Underwater Vehicles (AUVs) and underwater gliders are used for these applications. For underwater missions, there are number of limitations must be taken into consideration like limited mission length, stringent non-economic motion constraints, and limited communication with each other or with the home base. In order to support vehicle kinematic constraints, vehicle paths must satisfy geometric continuity [24-25] because of non- economic motion constraint requires that the vehicle motion must be a long smooth curvature without reverse it’s direction. In order to achieve G1 continuity and get shortest path length for point to point path planning, Dubins curves are used [26-27]. Also for unmanned aerial vehicle, Dubins curve proposed by using linear interpolation [28]. The multi-target multi-AUV task assignment and path planning problem is can be represent as multiple traveling salesperson (MTSP) problem. This problem can be solved by many methods like k-means clustering method [29], genetic algorithm [30-31], and heuristic search algorithm [32-33].

6. CONCLUSION AND FUTER WORKS

Many researches on robot path planning in a three dimensional environment have concentrated. This paper presents a survey on a three dimensional path planning algorithms. It dependents on the static and dynamic obstacles when the mobile robot draw it’s trajectory to the goal. Static obstacle avoidance is a way of avoiding obstacles that is static in shape and position and dynamic obstacle avoidance is a way of avoiding obstacles that is dynamic in position. Also this paper discusses the type of the three dimensional vehicles: UAV and AUV. The UAV works as a flying robot and the UAV works as a swimming robot. Two types of data structure are discussed: the Voxel grid and the volumetric navigation mesh. In Voxel grid the data structures organize their data in a tree-based way; so that actions like searching, adding or removing objects can have a good average time complexity where the volumetric navigation mesh approach is mainly limited to modeling the walk-able areas.

7. ACKNOWLEDGMENTS

Big thanks to the experts who have contributed towards development of the template.

8. REFERENCES

[1] A. T. Rashid, M. Frasca, A, A, Ali, A. Rizzo and L. Foruna,”Multi-robot localization and orientation estimation using robotic cluster matching algorithm”. Robotics and Autonomous Systems, Vol. 63, p.p. 108-121, 2015.

[2] O. A. Hasan, A. T. Rashid, R. S. Ali and J. Kosha,” A Practical Performance Analysis of Low-Cost Sensors for Indoor Localization of Multi-Node Systems “, Internet Technologies and Applications (ITA), Wrexham UK, September 2017.

[3] O. A. Hasan, A. T. Rashid and R. S. Ali,” Centralized approach for multi-node localization and identification “, Iraq J. Electrical and Electronic Engineering, Vol.12 No. 2, pp. 178-187, 20 1 6.

[4] O. A. Hasan, A. T. Rashid and R. S. Ali,” A Hybrid approach for multi-node localization and Identification “, Basra Journal for Engineering Sciences, vol. 16, no. 2, pp. 11- 20, 2016.

[5] A. T. Rashid, W. H. Zayer and M. T. Rashid,” Design and Implementation of Locations Matching Algorithm for Multi-Object Recognition and Localization”, Iraqi Journal of Electrical and Electronic Engineering, Vol. 14, No. 1, p.p. 10-21, 2018.

[6] A. T. Rashid, A. A. Ali, M. Frasca and L. Fortuna, " Path planning with obstacle avoidance based on visibility binary tree algorithm," Robotics and Autonomous Systems, vol. 61, pp. 1440–1449, 2013.

[7] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " An algorithm for Path planning with polygon obstacles avoidance based on the virtual circle tangents", Iraq J. Electrical and Electronic Engineering, Vol. 12, No. 2, pp. 221-234 , 2016.

[8] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " Prediction-Based Path Planning with Obstacle Avoidance in Dynamic Target Environment ", Basra Journal for Engineering Sciences, Vol. 16, No. 2, pp. 48 – 60, 2017.

[9] M. Montemerlo, S. Thrun, D. Koller, and B. Webfeet, “Fast SLAM: A factored solution to the simultaneous localization and mapping problem,” Proceedings of the National conference on Artificial Intelligence, 2002, pp. 593–598.

[10] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, “The office marathon: Robust navigation in an indoor office environment,” in IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 300–307.

[11] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, “Autonomous driving in urban environments: approaches, lessons and challenges,” Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 368, no. 1928, pp. 4649–4672, 2010.

[12] R.He, S.Prentice, and N. Roy, “Planning in information space for a quadrotor helicopter in a GPS-denied environment,” in IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 1814–1820.

[13] J. Tisdale, Z. Kim, and J. Hedrick, "Autonomous UAV path planning and estimation," IEEE Robotics & Automation Magazine, vol. 16, no. 2, pp. 35–42, 2009.

[14] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," IEEE Trans. On Robotics, vol. 23, no. 2, pp. 331–341, 2007.

[15] L.Kravaki, P.Svestka, J.C.Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[16] S. M.LaValle and J.J.Kuffner,"Randomized kinodynamic planning," International Journal of Robotics Research, vol. 20, no. 5, pp. 378– 400, 2001.

[17] S. Karaman and E.Frazzoli,"Incremental sampling-based algorithms for optimal motion planning," International Journal of Robotics Research, vol. 30, no. 7, pp. 846–894, 2011.

[18] E. Stumm, A. Breitenmoser, F. Pomerleau, C. Pradalier, and R. Siegwart,"Tensor-voting-based navigation for robotic inspection of 3D surfaces using lidar point clouds," The International Journal of Robotics Research, vol. 31, no. 12, pp. 1465–1488, Nov. 2012.

[19] Waveren , Rothkrantz 2006.

[20] Edward Angel and Dave Shreiner. Interactive computer graphics: A Top-Down Approach with Shader-Based OpenGL. Pearson Education Limited, 2012.

[21] Samuli Laine and Tero Karras. E_cient sparse voxel octrees{analysis, extensions, and implementation. NVIDIA Corporation, 2, 2010.

[22] Wouter G van Toll, Atlas F Cook IV, and Roland Geraerts. Multi-layered navigation meshes, 2011.

[23] Chyba,M.,Haberkorn,T.,Smith,R.N.,Choi,S.K.,2008.Desi gnandimplementation of time efficient trajectories for autonomous underwater vehicles. Ocean Eng. 35 (1),63–76.

[24] BesadaPortasE.,DeLaTorre,L.,DeLaCruz,J.M.,DeAndrés -Toro,B.,2010. Evolutionary trajectory planner for multiple UAV sin realistic scenarios. IEEE Trans. Robot.26(4),619–634.

[25] Roberge,V.,Tarbouchi,M.,Labonte,G.,2013.Comparisono fparallelgenetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Ind.Inf.9(1),132–141.

[26] Alvarez,A.,Caiti,A.,Onken,R.,2004.Evolutionary path planning for autonomous under water vehicles in a variable ocean. IEEE J.Ocean.Eng.29(2),418–429.

[27] Zheng,C.,Li,L.,Xu,F.,Sun,F.,Ding,M.,2005. Evolutionary route planner for unmanned air vehicles. IEEE Trans. Robot. 21 (4), 609–620.

[28] Nikolos,I.K.,Valavanis,K.P.,Tsourveloudis,N.C.,Kostara s,A.N.,2003.Evolutionary algorithm based off line/online path planner for UAV navigation. IEEE Trans. Syst. Man Cybern. BCybern.33(6),898–912.

[29] Hollinger, G.A.; Choudhary, S.; Qarabaqi, P.; Murphy, C.; Mitra, U.; Sukhatme, G.S.; Stojanovic, M.; Singh, H.;Hover, F. Underwater Data Collection Using Robotic Sensor Networks. IEEE J. Sel. Areas Commun. 2012,30, 899–911.

[30] Liu, L.; Liu, Y. On Exploring Data Forwarding Problem in Opportunistic Underwater Sensor Network Using Mobility-Irregular Vehicles. IEEE Trans. Veh. Technol. 2015, 64, 4712–4727.

[31] Smith, R.N.; Huynh, V.T. Controlling Buoyancy-Driven Profiling Floats for Applications in Ocean Observation. IEEE J. Ocean. Eng. 2014, 39, 571–586.

[32] Garau, B.; Alvarez, A.; Oliver, G. Path Planning of Autonomous Underwater Vehicles in Current Fields with Complex Spatial Variability: An A* Approach. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Shatin, China, 18–22 April 2005; pp. 194–198.

[33] Sun, B.; Zhu, D. Three dimensional D*Lite path planning for Autonomous Underwater Vehicle under partly unknown environment. In Proceedings of the 2016 12thWorld Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 3248–3252.