.

EFFICIENT CAMERA SELECTION FOR MAXIMIZED TARGET COVERAGE IN UNDERWATER ACOUSTIC SENSOR NETWORKS

by

Abdullah Albuali

B.S., King Faisal University, 2009

A Thesis
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

Department of Computer Science
in the Graduate School
Southern Illinois University Carbondale

December 2014

THESIS APPROVAL

EFFICIENT CAMERA SELECTION FOR MAXIMIZED TARGET COVERAGE IN UNDERWATER ACOUSTIC SENSOR NETWORKS

By

Abdullah Albuali

A Thesis Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Computer Science

Approved by:

Kemal Akkaya, Chair

Henry Hexmoor

Michael Wainer

Graduate School
Southern Illinois University Carbondale
October 31st, 2014

AN ABSTRACT OF THE THESIS OF

ABDULLAH ALBUALI, for the Master of Science degree in Computer Science, presented on 31 October 2014, at Southern Illinois University Carbondale.

TITLE: EFFICIENT CAMERA SELECTION FOR MAXIMIZED TARGET COVERAGE IN UNDERWATER ACOUSTIC SENSOR NETWORKS

MAJOR PROFESSOR: Dr. Kemal Akkaya

In Underwater Acoustic Sensor Networks (UWASNs), cameras have recently been deployed for enhanced monitoring. However, their use has faced several obstacles. Since video capturing and processing consume significant amounts of camera battery power, they are kept in sleep mode and activated only when ultrasonic sensors detect a target. The present study proposes a camera relocation structure in UWASNs to maximize the coverage of detected targets with the least possible vertical camera movement. This approach determines the coverage of each acoustic sensor in advance by getting the most applicable cameras in terms of orientation and frustum of camera in 3-D that are covered by such sensors. Whenever a target is exposed, this information is then used and shared with other sensors that detected the same target. Compared to a flooding-based approach, experiment results indicate that this proposed solution can quickly capture the detected targets with the least camera movement.

ACKNOWLEDGMENTS

It is a pleasure to thank God and the many people who made this thesis possible.

First of all, it was a great honor to have had my M.S. studies under the direction of such an excellent advisor, Professor Kemal Akkaya. I would like to express my gratitude to him for all his invaluable guidance and support while conducting this research. His advice helped me overcome obstacles during every step of writing this thesis, and I am grateful for his patience and wisdom.

My sincere thanks go to my other committee members as well, Professor Henry Hexmoor and Professor Michael Wainer, for helping me with their thoughtful suggestions and insightful comments. It was also a great pleasure to work with Dr. Fatih Senel. He is truly one of a kind and a great professor. His personality always has a special impact on me. He helped me encouraged, find solutions when it looked like it is the end, and for this I thank him very much. My countless thanks go to Dr. Jerzy for his ceaseless support with the mathematical formulas that were central to analyzing my results. I would have been lost without his assistance.

I would also like to thank my friends, Izzet F. Senturk, Mahbubul Arefin Khan, and Abdulrahman Al-Ahmadi for their support and helpful comments. I also wish to thank my friends, Saif, Ervin, and Nikhil who stood by me and helped with technical aspects of my thesis.

Finally, to the best person in my life, my mother, who always supported me and encouraged me to do better. And to my father, who was always there for me when I needed him the most. To my lovely wife, Tahany, and my son, Abdulrahman, who were with me in all the hard times. And to all my family members. To them, this thesis is dedicated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Underwater Acoustic Sensor Networks (UWASNs) can consist of a large number of sensors above and underwater that communicate via acoustic links [1][2]. Similar to terrestrial wireless sensor networks (WSNs), UWASNs provide advantages in terms of coverage quality, labor, cost and deployment, as opposed to traditional underwater sensor networks. In the previous decade, numerous studies on the issues related to communication underwater have shown that RF signals do not work well underwater. The design of acoustic modems, channel modeling, medium access, and sensing and routing issues have thus been the main focus of recent research [3][4].

Battery-operated cameras in underwater sensor networks can also be deployed in addition to regular acoustic sensors to perform object tracking and identification via the image/video of the object detected. By using mechanisms such as pumps [5], both sensors and cameras can move vertically. However, as the processing and transmission of multimedia data consumes an enormous amount of battery power, cameras could be set on sleep most of the time and only used on demand [3]. For example, an ultrasonic sensor [6] can detect a target and then activate a camera to capture the target for a certain period of time. On the other hand, if there are not enough cameras close to the target (due to random deployment), the cameras may not be able to capture the target. A possible solution to this problem is to adjust the depth of the camera to be closer to the target. Subsequently, the challenge becomes distributing the cameras to have maximum coverage of the target with minimum overlap. The location and orientation of the cameras are thus of central concern.

While finding the right location is crucial in calibrating the cost of the depth adjustment, camera orientation will be used to determine whether there are any overlaps or if the object detected is fully covered. The aim of the present study was to reduce the total energy cost (i.e., distance) that stems from depth adjustment while at the same time assuring full coverage or maximizing coverage of the designated target. In brief, instead of having full coverage at the cost of many overlaps it is preferable to find an alternative solution that offers full coverage with fewer overlaps and less energy consumption.

Another important challenge is communication delay. Selecting 1-hope cameras is a good approach. The transmitted information of cameras within multiple hops may not make reaction time sufficient to capture the target. For example, there is a chance that the target will leave the camera's field of view if it is moving fast while camera is still being altered and moved to the right depth. Additionally, one of the biggest issues, particularly for UWSNs, is propagation delays being higher most of the time. Consequently, the problem of the present study has the additional constraint of the time that needs to be taken into account when selecting the cameras. To the best of my knowledge, this problem is novel and has not been previously studied.

Connectivity is another constraint in UWASNs. When the camera adjust its depth to an optimal location, there is no guarantee that it will still be connected to rest of the network. The captured data will be transmitted directly if such a camera cannot communicate with any of the nearby sensors. The camera would then need to move to another location to reestablish connectivity with a sensor, which may result in its failing to capture the target. Therefore, it makes more sense to choose a location that will also guarantee a link with a sensor. Note that this study proposes a UWSN that is already connected by using such techniques as the one employed in [5].

This thesis is organized as follows: In the next chapter, we present the literature review. Chapter III describes different elements settings of the proposed underwater wireless acoustic sensor networks. These settings are representing our system model: Camera setting , Acoustic sensor setting , and Target setting. Section IV represents camera selection process and shows different algorithms were used for that purpose. In chapter V, computation of coverage algorithm are introduced . Section VI then shows experiement results that show how our proposed solutions have overcome the coverge problem of target in UWASNs. That means closest cameras with least overlap and energy cost will be actuated. Conclusions and futur works are finally drawn in Section VII.

CHAPTER 2

LITERATURE REVIEW

This chapter summarizes the current state of the field, dividing it into two subsections: 1) Coverage and Connectivity in UWSNs and 2) Camera Coverage.

2.1 COVERAGE AND CONNECTIVITY IN UWASNS

Achieving full coverage and connectivity in UWASNs assuming both random and manual node deployment has been thoroughly studied in the past [7][8]. For instance, the main goal in [7] was to analyze the implications of sensing and communication ranges for coverage, connectivity and network diameter in a UWASN. The author established conditions for the node transmission range required for achieving a degree of connectivity and coverage.

Alam and Haas [8] investigated the problem of achieving maximal 3-D coverage with the least number of sensors. As opposed to [7], random deployment was not used as an option in this case. Rather, manual deployment was considered for determining the least number of sensors needed. The authors argued that space filling polyhedrons would be more suitable for 3-D applications. Note that these are the optimal solutions when the number of sensors and the coordinates of the area are known to a central authority. Although they claimed at the end of their paper that the algorithms could be run in a distributed manner by selecting a leader, it required that each node reach every node and that the nodes be capable of being moved to the desired location in 3-D space. They stated that this was a problem in 3-D and left it as a concern for future studies.

In addition to the abovementioned research, coverage improvement has also been heavily studied in both a distributed and centralized manner [9][10][11]. Pompili et al. [9], used the bounds derived in [7] to validate the effectiveness of their random node deployment scheme for UWASNs. Sensors were deployed at the bottom of the ocean along with a few gateway nodes. The sensors sent their data to nearby gateways, which forwarded the data over vertical communication links to floating buoys on the surface. The idea was to adjust the depths of the sensors after their deployment at the bottom of the ocean to provide 1-coverage. The initial deployment was random and based on a grid. However, the deployment was controlled by a central station that told each sensor where to go after its initial positioning to achieve 1-coverage. Thus, the approach was not fully distributed.

In [10], the nodes adjusted their levels based on coverage overlap with their neighbors. To determine the sensor coverage overlaps, nodes were clustered and links defined between any two nodes if there were any overlap. Graph coloring was then applied within each group, and for each color a new depth was calculated. After this stage, any possible overlaps among the nodes in different groups were eliminated. This improved the coverage significantly but did not guarantee connectivity. It was claimed that only within a certain transmission and sensor range ratio could connectivity also be ensured. While this has been shown to be the case with a certain number of nodes and ratios, the approach cannot guarantee the connectivity in all cases. [11] also proposed a distributed algorithm for improving coverage in UWASNs. As in the present study's setup, the nodes are deployed randomly at the top and connected to buoys with wires. After this initial deployment, the nodes adjust their depths to improve coverage.

UWASNs can be either stationary, where sensor nodes are anchored to the ocean floor or surface buoys, or they can be free floating. In the second case, although some nodes can be propelled and navigate underwater, due to cost constraints, sensor nodes usually float freely and drift with the motion of the aquatic environment. Nodes with navigation capability are known as Underwater Autonomous Vehicles (UAVs) and may reside in the network for localization [12], data collection or similar purposes [13]. The present study considered free floating sensor nodes submerged underwater to depths of several hundred meters to provide measurements from an oceanic environment. In such mobile UWSANs, nodes move with the force of the surface winds and underwater currents. In the literature, there are several mobility models for UWASNs and those models are significantly different from the motion of nodes in terrestrial Mobile Ad Hoc Networks (MANETs). For a detailed review of those models the reader is referred to [14]. Among the few UWASN-related mobility models, in [15] the authors proposed a tidal current model. According to this model, the motion of the sensor nodes is governed by tidal changes. This model applies to coastal environments. In [16], the authors modeled the ocean currents as layers with equal thickness and varying speeds, and assumed that the sensor nodes would move with those currents. This model presents group mobility for UWASNs; however in practice, there needs to be non-negligible correlation between the layers. A widely-referenced underwater mobility model is the meandering current mobility model (MCM), which relies on mathematic models of the ocean currents [17]. The initial version of MCM considers sub-surface behavior while an extended version includes surface mobility in [18]. The random motion resulting from the surface winds is combined with the sub-surface current-induced motion to determine the mobility of the UWASN.

2.2 CAMERA COVERAGE

There are only a very few works that deal with Camera coverage in underwater acoustic sensor network for target coverage. For example, in [19], the authors mentioned that in order to get 3-D image or a video, then, there must be at least two cameras are parallel with each other to maintain the full coverage of such object. This is only to give idea about why one camera is not sufficient to have full coverage of 3-D target as in our approach. Also, one study is meant to target how possibly easy to use and distribute Wireless Camera sensors in vast environment with a huge number of sensors from couple of hundreds to thousand depends on field of interest, due to the low cost of sensors [20].

In paper [21], authors give an idea about how to calculate Field of View of any camera in 2-D. However, [22] shows exaclty how to calculate FOV of camera in 3-D manner. The last one is the setting we went with in our approach.

There are many works proposing different solutions about surveillance system and regarding monitoring areas of interest [23] . However, [24] is to get WMSN to be an efficient by looking for object-related phenomena instead of geometrical one relationship. Also, some of works talked further to how make sure monitoring area controlled by very high-quality video streaming while maintain some constraints. Authors in [25] , for example, designed a cross-layer system to enhance video quality for Wireless Multimedia sensor. That can be a remarkable enhancement of Wireless Multimedia sensor researches.

Most of works are done in 2-D environment. For instant, in [26], authors proposed a solution that may apply for traditional WSN that will integrate the Camera coverage and routing problem to be based on video networks. They suggested to use a special way that how routing should designed to be more scalable to video-based wireless network. Also, [27] [28] and[29], they mentioned different challenges of WSN and some solutions about processing time, and coverage. In addition, some papers talked about partial coverage of target by participated cameras based on energy consumption per each [30], this approach is not belong to our work since we are looking for full coverage of target.

Talking about 3-D environment, [31][32] attend to produce a full 3-D representation of the observed environment to improve its exploration and analysis by a collaboration among multi-Wireless Camera sensors. However, this will not going to be necessary for our approach since we set cameras sensors to be in sleep till it is actuated by detected acoustic sensors to capture a specific target. One of the closest work to ours have been reported in [33] [34] , and [35] . Whereas [36] is doing maximum coverage in 2-D, the three first is working in 3-D environment. While, all of these approaches consider minimizing FoV of cameras by eliminating redundant data among them, our approach is consider the target's critical points to help topology reducing the the overlapping among actuated cameras by our robust algorithm. In addition, since all of these approaches did not mention if target is located in area that uncovered. In contrast, our approach guarantee that uncovered positions of target can be reachable. Of course, there is a sacrifice on energy while camera moves. Thus, with help of our algorithm of coverage (i.e., discussed in Chapter 5) this problem can be minimized by selecting only cameras that have higher coverage while have minimum distance to its best displacement close to target's scene.

Since there are not much papers about calculation of distance between camera and the object in Wireless Acoustic sensors network. In fact, there is a study that embedded laser pointer on the UAV and use mathematic model in order to calculate the distance between cameras or between cameras and object [37]. However, this approach required using UAVs which is out of our scope in this thesis. Instead, with help of 1-hope sensors, locations of cameras can be calculated with help of a specific mathematical model. Thus, it will tell how much exactly the distance between any location of camera and location of the target.

CHAPTER 3

BACKGROUND ON UNDERWATER WIRELESS ACOUSTIC SENSOR NETWORKS

3.1 PRELIMINARIES

A UWASN in the present study is a set of acoustic sensors and multimedia sensors (e.g., GX support OEM [6] and CmuCam3 [38]). When a target is detected by acoustic sensors, it will be captured or shot on video by multimedia sensors as needed. In this study, multimedia sensors are hereafter referred to as cameras. An example of a UWASN is shown in Figure 3.1.

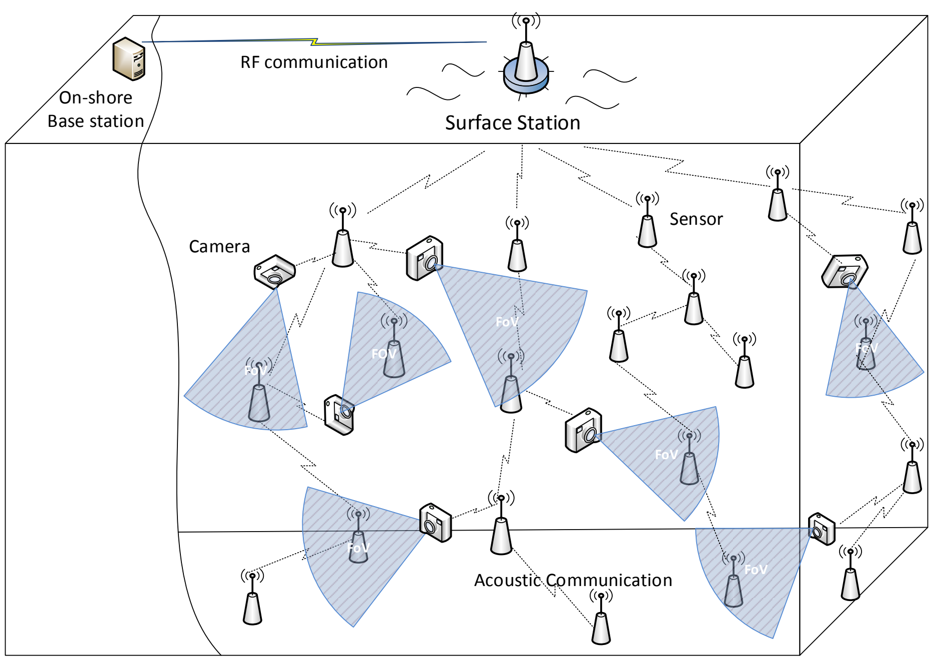

Figure 3.1. A 3-D UWASN with cameras and sensors transmitting data to a surface station

Total nodes of such a UWASN topology is represented by n, which consists of a fixed random number of acoustic sensors m and cameras k. The acoustic sensors m are supposed to appear in a spherical shape with a fixed random transmission range. As mentioned before, in order to fully cover or maximize coverage of a target, two variables should be considered: Camera Structure, and the Object (i.e., Target) structure that both play important parts in this study's approach. To better understand these variables, camera setting is explained first.

3.2 CAMERA SETTING

In Wireless Multimedia Sensor Networks (WMSNs)[39], each camera k has a specific field-of-view (FOV) w and depth-of-field (DOF) s, which are the angle and the distance respectively where the camera can capture an accurate image/video in 2-D as seen in Figure 3.2.

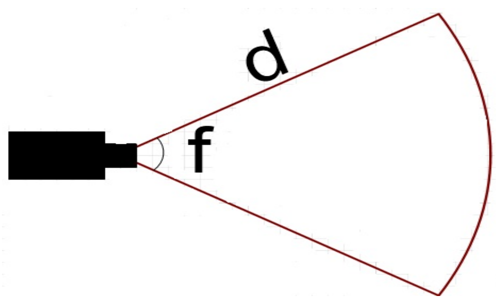

Figure 3.2. (2-D) Camera Model

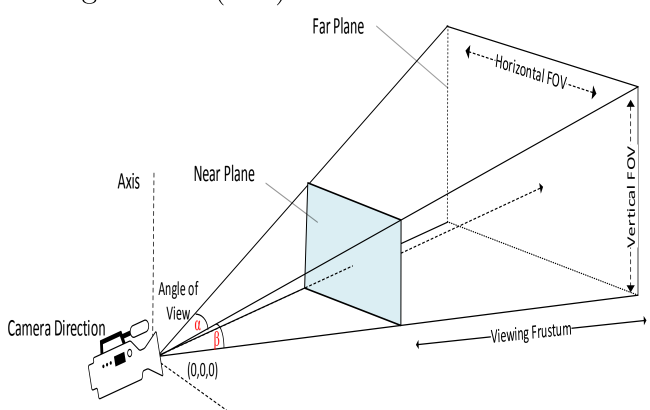

Figure 3.3. (3-D) Camera Model

Each camera has a certain transmission range to communicate with other nodes. This transmission range is assumed to be larger than or equal to DOF. The cameras are assumed to have a fixed random position and orientation (i.e., the direction the camera faces). While the exact boundaries of the monitored area are known, the events (i.e., their location, shape and radius) are not known in advance and are assumed to appear randomly within the monitored area. Again, this network is meant to appear in 2-D.

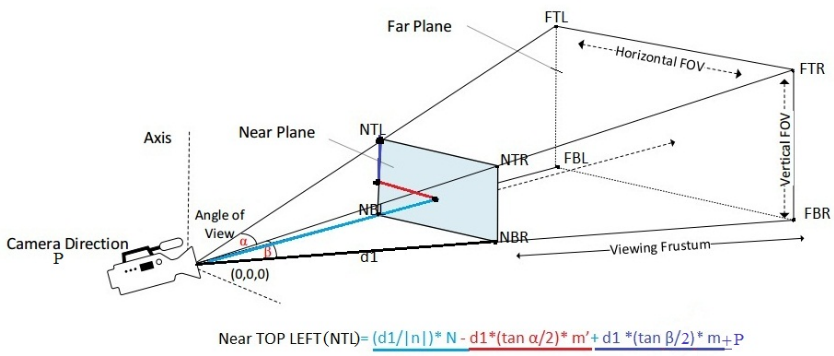

Figure 3.4. Calculating First of Eight Critical Points of Camera Frustum: (NTL)

In a UWASN (i.e., in the present approach), however, camera sensors are used in order to capture any target of interest in 3-D. These cameras, in particular, have a different definition from other WMSNs as explained above. For example, in a terrestrial environment, a camera can shoot a video or take pictures of a target in 2-D. In a UWASN 3-D environment[22], however, it is necessary to have a camera with a FOV structure consisting of two angles: Horizontal FOV and Vertical FOV, as we are going to explain later (see Figure 3.3). As in WMSNs, each camera in UWASNs can communicate with other nodes via a specific transmission range that can be equal to or larger than the DOF. This feature and its impact upon the present approach is explained in greater detail in Chapter 4. The cameras are supposed to have an exact random orientation (i.e., where the camera faces) and location, and they can only move vertically (i.e., Z-axis of camera) to adjust their depth to maximize the coverage of the target.

Setting up the camera involves calibrating a Camera Frustum (see Figure 3.4). In order to do that, the eight coordinates of the near and far end planes of the camera should be calculated. These eight 3-D coordinates are computed based on the location, orientation and FOV of the camera with help from the variables in Table 3.1. These variables are used in specific equations (i.e. equations [3.1-3.8]), where each line is responsible for calculating one coordinate out of the eight that represent camera frustum. Table 3.1 contains all values used for checking camera coverage. NTL, NDL, NTR and NDR are the closest plane of a camera. By knowing them, any point located before this plane will not be seen. Far plane, on the other hand, is the farthest point a camera can see and consists of FTL, FDL, FTR and FDR, and any point after this plane will not be seen as well.

Horizontal FOV α and Vertical FOV β combined can determine how much a camera's angles can be opened. In general, the FOV can be defined as the angle of the triangle in the horizontal or vertical plane of the camera. The α has many values with the width of the picture plane, and β varies with the height of the picture plane. A wide variety of cameras models and their relative FOVs are shown in Table 3.2.

$$NTL = (d_1/|n|) * N - d_1 * (\tan\alpha/2) * m' + d_1 * (\tan\beta/2) * m + P \tag{3.1}$$

$$NDL = (d_1/|n|) * N - d_1 * (\tan\alpha/2) * m' - d_1 * (\tan\beta/2) * m + P \tag{3.2}$$

$$NTR = (d_1/|n|) * N + d_1 * (\tan\alpha/2) * m' + d_1 * (\tan\beta/2) * m + P \tag{3.3}$$

$$NDR = (d_1/|n|) * N + d_1 * (\tan\alpha/2) * m' - d_1 * (\tan\beta/2) * m + P \tag{3.4}$$

$$FTL = (d_2/|n|) * N - d_2 * (\tan\alpha/2) * m' + d_2 * (\tan\beta/2) * m + P \tag{3.5}$$

$$FDL = (d_2/|n|) * N - d_2 * (\tan\alpha/2) * m' - d_2 * (\tan\beta/2) * m + P \tag{3.6}$$

$$FTR = (d_2/|n|) * N + d_2 * (\tan\alpha/2) * m' + d_2 * (\tan\beta/2) * m + P \tag{3.7}$$

$$FDR = (d_2/|n|) * N + d_2 * (\tan\alpha/2) * m' - d_2 * (\tan\beta/2) * m + P \tag{3.8}$$

Table 3.1. Variables of Camera Setting Coverage

Variable	Description
NTL, NDL, NTR and NDR	Near plane as: Near Top Left point, Near Down Left point, Near Top Right point and Near Down Right point
FTL, FDL, FTR and FDR	Far Top Left point, Far Down left, Far Top Right and Far Down Right
d_1	Distance from Camera position P to Near Plan
P	Coordinate of Camera position
d_2	Distance from Camera position P to Far Plan
$v = [x, y, z]$	Vector of the position of the object
α (alpha)	The angle camera captures image across "horizontally", namely Horizontal FOV
β (beta)	The angle camera captures image "vertically", namely Vertical FOV
$n = [n1, n2, n3]$	Vector indicating the direction the camera is looking at. Assume that n is a unit vector (unit length)
$m = [m1, m2, m3]$	Unit vector indicating the "vertical" orientation of the camera. Clearly, m $\perp$ n (m is perpendicular to n)
m'	Same for the "horizontal" axis. May be calculated using cross product, namely $m' = m \times n$

Table 3.2. Underwater Camera Setting from different manufacturers

Camera Type	Underwater Camera 1 Fish finder- Monitoring [40]	Underwater Camera 2 wide-angle for Monitoring [41]	Underwater Camera 3 wide-angle for identification [42]
Horizontal FOV	92 Degree	360 Degree	360 Degree
Vertical FOV	69.2 Degree	180 Degree	180 Degree
Near plane (d1) Dist. Day/Night	0.15m	0.15m	0.15m
Far plane (d2) Dist. Day/Night	6-10m/2-5m	6-10m/5-6m	3-5m/ 2-3m
Number of Leds IR/white	18 LEDS	12 LEDS	6-8 LEDS

Equations [3.1-3.8] are used to set up Camera Frustum. Any point within the 3-D area can be checked by the cameras. Parameters like Camera orientation n, and both m, and m' are , also, being used in calculation the critical points. In order to get values of m and m', we built a mathematical model to get these values, see equations 3.9 and 3.10.

$$n = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \Rightarrow m = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} -\frac{ac}{\sqrt{a^2+b^2}} \\ -\frac{bc}{\sqrt{a^2+b^2}} \\ \sqrt{a^2+b^2} \end{bmatrix} \tag{3.9}$$

$$m' = \begin{bmatrix} b\sqrt{a^2+b^2} - \frac{bc^2}{\sqrt{a^2+b^2}} \\ -a\sqrt{a^2+b^2} - \frac{ac^2}{\sqrt{a^2+b^2}} \\ 0 \end{bmatrix} \tag{3.10}$$

Where we derived m, the vertical vector orientation value from n, and m', the horizontal vector orientation value from m.

All of these steps are crucial factors to build up camera structure in the 3-D underwater environment. Thus, it will help by checking any target in the monitoring area that is fully or partially covered. The idea of detecting such critical points of the target was inspired by [31] [43] and considers the target to be fully covered if these points are captured by such 3-D cameras. However, this information and related topics will be discussed in detail in chapters 4 and 5.

3.3 ACOUSTIC SENSOR SETTINGS

Figure 3.5. Acoustic Sensor Sample

Before talking about target settings below, let us take a brief overview of acoustic sensor settings that are used underwater. Multimedia sensors (e.g., GX support OEM [6]) are used in the present study as acoustic sensors. These sensors have a spherical shape to detect any target passes through their sensing range. Figure 3.5 shows an ultrasonic sensor that is used underwater. It has a sensor range of 10 meters and is how the topology can be arranged accordingly. Usually, acoustic sensors build most of the topology underwater since they work as detector nodes responsible for sending, receiving, processing, and calculating results. They are deployed to perform collaborative monitoring. This study used connected topology called connected dominating set, or (CDS) which was well proposed in [5]. In general, since the topology is a distributed approach, communication between nodes is within 1-hope node for energy-saving purpose.

3.4 TARGET SETTINGS

The target in a 3-D underwater environment in this study appears in a 3-D shape, i.e., as a rectangular prism. This can become challenging for such cameras to maximize the coverage of a 3-D shape underwater. The target is represented as a rectangular prism, or cubiod, as seen in Figure 3.6. The use of these critical points was inspired by [44] in which the system used keypoints of the ASIFT algorithm [45] to measure the coverage.

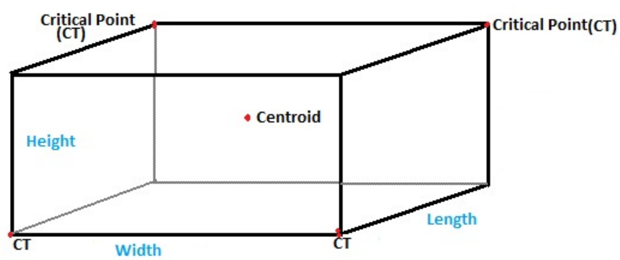

Figure 3.6. Target Model: 5 Critical Points of Target

In UWSNs, a 3-D target is usually considered fully covered if the FOV of any set of the 3-D-based cameras covers all of that target's critical points, regardless of their orientation. In practical, we decided to represent the prism with Five points: two points for each edge (i.e., if we considering a rectangle, these two points will be two corresponding diagonals), and the centroid of the prism which is the 3-D coordinate of center point of rectangular prism. In fact, with help of centriod point , we could calculate the other critical four points.

This approach leads to the following question: how can these points be determined and defined? These points are to be calculated once the acoustic sensors m detect a target. By knowing its position (i.e., target point, which is the centroid in this case), the other four critical points can be calculated from that point.

Note that it is logically impossible for one camera in a 3-D environment to capture a

3-D object alone at one time. Thus, at least two cameras are needed to obtain the full coverage of such a target [19]. Therefore, the best case scenario in this approach usually involves the use of many cameras to cover the object. The object can also be various sizes. That means that during testing multiple object sizes were considered in different scenarios with different results. The testing process is explained in greater detail in Chapter 5.

CHAPTER 4

CAMERA SELECTION

This chapter describes the camera determination problem in UWASNs and presents three algorithms for camera determination:

- Camera Database Creation Algorithm
- Closest Camera Search Algorithm
- Camera Selection Algorithm

At the same time, this approach avoids or else overcomes most of the disadvantages of previous approaches.

4.1 PROBLEM DEFINITION

The problem of the present study can be formally defined as follows: ”Given a connected network of size n sensors and m cameras where $n >> m$ and a surface gateway, if there is a mobile target passing from the monitoring area, the goal was to detect it first with ultrasonic sensors and then get a video of this target to understand its category (i.e., a fish or submarine). Since the cameras were on sleep mode and not available everywhere in the region, the goal was to maximize video coverage of the target with minimum energy cost (the movement distance for the cameras) and delay by using a distributed approach.” This chapter focuses on deployment level of the system and ways to save the information of all cameras in the network by using 1-hope sensors, searching for the best cameras, and at the end, how to determine the best cameras in the network in terms of distance and coverage.

4.2 APPROACH OVERVIEW

Our approach was based on having the cameras cover the bounded area after determining the possible boundaries of the target, which is always based on the locations of the sensors that detected the target. The computations for overlaps are done in advance and thus the cameras move only when the computations are finished. There are two steps: 1) each sensor has a spherical sensing range for target detection. Therefore, when they detect a target, the target is wholly or partially within this sensing range and will potentially be within this range even if the target moves further away. As a result, if we would like to capture the target, we need to capture the spherical sensing region of a sensor with the camera(s). In this step, each sensor will determine the closest cameras which will maximize the coverage of its spherical region. This needs to be done in advance. 2) When multiple sensors detect a target, the target should be within the intersection region of these coverage ranges. The sensors should check possible overlaps and decide which cameras to bring. Basically, any camera which does not add any significant coverage is discarding from the list.

4.2.1 Proactive Determination of Cameras

Each sensor determines the closest camera to itself and its coverage based on the FOV and DOF parameters as well as the location of the cameras. This is a step which needs to be done in advance when the network is deployed. The computation needs to determine the exact location of the camera to know where it will be relocated. Obviously, the goal is to maximize the coverage (full coverage is desirable). However, the camera, when relocated, needs to be within transmission distance r to one of the sensors in the network to have connectivity with the rest of the nodes. There are two possible situations in this case:

- The camera can be within the transmission range of the sensor that detected the target. The computation can be done at the sensor.

- The camera can be within the transmission range of any of the sensors in the network as long as it covers the sensor's sphere (assuming DOF is greater than the transmission range). This is not easy to determine since the sensor will not know the location of every other sensor and the topology of the network. Obviously, this is because this approach runs on a distributed topology.

This study followed the first case by assuming that DOF was less than the transmission range. The search process is explained below.

4.2.1.1 Search Options

We need to check all cameras that will be within a distance of r from the sensor in 2-D (x-y). This is because, if this distance is greater than r, then even though the depth is adjusted, the camera and sensor will not be able to communicate.

The challenge is whether to flood the network to gather this information (like route discovery) or to use CDS to handle it. If the former is used, then this will create a flooding for each sensor, which is very inefficient. Note that a reactive mechanism can be used when needed but it will be very slow due to longer propagation delays in UWSNs. Thus, as the camera moves to a new depth, the target will probably be gone. Due to this problem, we eliminate that solution. If the latter is used, then each dominator should keep a hash table of cameras along with their locations and provide this information to the sensor that asked for it. The space may be an issue but we can handle it by keeping the storage local to the neighborhood.

4.2.1.2 Search Algorithm

We decided to follow the CDS-based approach. The network will determine its CDS and thus at the end each camera will be either a dominator or will have a link with a dominator node to talk to. To keep the storage overhead low, each dominator will only maintain the cameras that belong to their 1-hop neighbors (and their own). The motivation stems from the fact that in most cases, the cameras that will need to be relocated will be from the neighborhood and thus 1-hop distance may already meet this condition in a majority of cases. The tradeoff between the size of the neighborhood and the quality of the solution in terms of coverage will be investigated in the experiments.

To start with, the camera database algorithm is explained as follows:

Algorithm 1 cameradatabase

1: Each camera c_i reports its loc and orientation to its d= dom(c_i)

2: Each dominator maintains a list of cameras called *camlist* = (id, loc, orientation)

3: Each dominator will broadcast this *camlist* to 1-hop dominators

4: Each dominator will update its *camlist*

Algorithm 2 closesestcamerasearch(i)

1: When a sensor s_i detects a moving target, it sends a DETECTED message to d_i = dom(s_i)

2: **for all** c_j in *camlist* of d_i **do**

3:     check cov_ratio of c_j on s_i's sphere by computing the exact location for c_j

4:     if full coverage of s_i is provided, exit

5: **end for**

6: **if** no camera exists **then**

7:     Send NONE message to s_i

8: **else**

9:     Send (id, loc, orientation) of c_j to s_i

10: **end if**

Algorithm 3 cameraselection

1: Each sensor s_i receives (id, loc, orientation, cov_ratio) from 1-hop neighbors.

2: s_i determines whether it will be the leader or not by comparing with cov_ratio of others.

3: **if** s_i is the leader **then**

4: Determine coverage overlaps with each sensor

5: Decide which sensors will bring the cameras

6: **else**

7: Wait for the message from the leader if not leader

8: **end if**

4.2.2 Selection of Cameras

Regarding the exchange of this information among sensors, if a sensor detects an activity, it broadcasts this to its 1-hop neighbors. If there are any other sensors detecting the same activity, then there is a possibility that they may bring the same camera or multiple cameras which may have overlapping FOVs. Therefore, they need to exchange the coverage information.

Using the camera information from its dominator, a sensor will broadcast the information about the camera(s) it will move closer to itself. The message broadcast will have (id, loc, orientation, cov_ratio) to cover the sensor. This process will be led by the sensor with the highest coverage. If there is tie for highest coverage, it will be broken by node ID. We will set a minimum threshold for the cov_ratio. If this ratio is met by a camera, the camera will be relocated. If the coverage is already provided by other cameras (i.e., the exclusive coverage it provides is less than the threshold), then the camera(s) corresponding to the sensor will not be relocated. The leader sensor will make the decisions for others and inform them of whether they must bring their closest cameras.

CHAPTER 5

COMPUTATION OF COVERAGE

This chapter explains how computations are performed for 3-D target coverage and how some of the drawbacks of similar approaches are avoided.

5.1 PROBLEM DEFINITION

The problem of coverage step in camera and object is described here. Our approach was to have a target fully covered. To be more specific, the best cameras close by the target were needed that could capture its five critical points (i.e., four critical corners and a centroid) fully covered or most of them are covered by the best nearby cameras are our goal. One more important factor in our approach was obtaining full coverage with the least possible overlapping between actuated cameras. As a consequence, there will be greater energy loss, which is of course not preferable. The goal was to find a list of the best cameras that fully or maximally cover the 3-D object with the least energy consumption and the least overlap.

5.2 APPROACH OVERVIEW

In order to start doing the real checking we have to check all active cameras in the topology that are selected by nearby detected sensors. Cameras will be actuated and their depths adjusted to be close enough to cover the designated target. To do that, we have to consider two points: the camera frustum that will check all five critical points, whether they are inside chamber or not. Once any of the critical points of an object is checked positively inside the chamber, we should check for which cameras have the best overall position, are closest to the target, and have the most of the target points. This is done even before the camera physically adjusts its depth. This step is consider as an advantage of this approach since it saves energy that camera might experince. The key solution is getting the position of the target's centriod, first. Then, it is easy to get the remaining critical points of a given object. In next section, we will explain two perspectives: steps of Camera Frustum Checking formulas, and an algorithm for scanning the target's critical points. Subsequently, camera will decide its best location namely, the best displacement, in which, it covers most of the target's five critical points.

5.2.1 Camera Frustum Checking Steps

When a camera is actuated and asked to go to a specific location to be able to capture an image/video of a target, one possible way to do this is by checking the following steps by which the critical points of the object positions will be checked and determine whether it is inside the "chamber" or not; if and only if these three conditions are satisfied (see Table 3.1) for reference and keywords:

itemi

1. $d_1 < \langle n, v \rangle < d_2$
2. $-\tan(\beta/2) < \langle v, m \rangle \ / \ \langle v, n \rangle < +\tan(\beta/2)$
3. $-\tan(\alpha/2) < \langle v, m' \rangle \ / \ \langle v, n \rangle < +\tan(\alpha/2)$

Where v is the vector of the object's critical points. Notice that, in order to consider any points of interest inside the camera , any location of target's points P_i should return positive for all the three conditions. First, P_i must be located in between d_1 and d_2. For that, we use dot product between vector n which represent orientation of camera and the targeted point P_i as we deonted it as v in equation. Second step , we have to make sure P_i is located between two vertical borderes with help of β and tan of Camera Frustum (i.e., upper plane and below plane). Last point , P_i must be located between two horizontal borders of camera frustum with help of α and tan of Camera Frustum (i.e., left plane and right plane) in order to capture such target's points. At the end, we have to check if these three conidtions are satsified, then consider this point as covered and check the next P_i. The selected cameras are going to be listed at each dominator's database and ready to scan the target, and then, adjust their depth to their best displacement.

5.2.2 Target Scanning and Coverage Algorithm

Once each of the activated cameras has been asked to check the object's critical points and report its result to 1-hope dominator, we need to understand couple of important points. To help know the mechanism , see algorithms 4 and 5. Once such s camera starts to adjust its depth to scan the target for coverage purpose, we have problem states that where exactly the camera should stop at? And ,therefore, marks that location as starting location of camera for scanning the target. To answer this optimization problem, we created a mathematical model to help camera to stop at specific location in which it can start scan the target, see equations [5.1, 5.2] for New Z-coordinate of camera and [5.3, 5.4] to calculate distance from the initial Z-coordinate of Camera and new Z-coordinate Camera points. Notice that this is not the best displacement of camera in which the camera covers the most of the target. Indeed, this is only to set where the moving camera should stop to begin scanning the object either from the upper Z position of Target or the lower one depends on the position of camera in the monitoring area .

$$CamZ_{NewUp} = -\sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2} \times \frac{c}{\sqrt{(a)^2 + (b)^2}} + ObjMaxZ \tag{5.1}$$

$$CamZ_{NewDown} = -\sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2} \times \frac{c}{\sqrt{(a)^2 + (b)^2}} + ObjMinZ \tag{5.2}$$

$$h_{Above} = -\sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2} \times \tan\gamma + ObjMaxZ - Zc \tag{5.3}$$

$$h_{Below} = -\sqrt{(X_o - X_c)^2 + (Y_o - Y_c)^2} \times \tan\gamma + ObjMinZ - Zc \tag{5.4}$$

Where the table 5.1 lists the parameters of Moving Camera Equations.

Table 5.1. Moving Camera equation parameters

$\sqrt{(Xo - Xc)^2 + (Yo - Yc)^2}$	Distance between the initial position of camera and new Z of camera.
$\tan\gamma$	tan angle of orientation of camera
Z_c	Z-Coordinate of initial position of Camera
$ObjMaxZ$,$ObjMinZ$	Upper or below Z-Coordinate of object
$CamZ_{NewUp}$,$CamZ_{NewDown}$	New Z postion of camera
h_{Above},h_{Below}	Distance between initial Z camera and New Z position .

Now, we need to explain the algorithms 4 and 5, which responable about scanning the object, let us name it here a prim for better understanding of the algorithms. These algorithms are worked by using each of activated cameras. o begin with , we start from the closest camera vertically from the prim. If the cameras are above the prism, we sort these cameras in increasing order according to the vertical distance to prism, check equations [5.3, 5.4] . In fact, the prism has two start points : top z-coordinate and bottom z-coordinate in which cameras start to check. Therefore, we need to calculate the new z-coordinate of these cameras to know exactly where they should adjust their depth to . To do that , algorithms used equation [5.1] if the camera is located above the prism , and used 5.2] if the camera is under the prism. For better coverage, we use Camera Frustum Centroid to help cover more of prism. Once camera z-coordinate arrived to its new position and close enough to either top or below Z-coordinate of prism, we adjust Camera Frustm Centriod to have same value of prism Maximum (i.e., upper) Z-coordinate or the Minimum (i.e., lower) one. For now, in a "for loop" start moving downwards the New Camera Frusutm z-coordinate, let us name it $NewCameraZ$ for better understanding, by x meters. At each iteration check if the camera covers any of these points. Assume at depth level , the camera $NewCameraZ$ covers p1 and p2. Mark these points p1 and p2 covered and remove them from list P. However, If cameras is under the prism, we do the same except by decreasing (moving upwards) $NewCameraZ$

by x meters. At each iteration check if the camera covers any of these points as It is explained in on the above case.

Note that at the end of this checking our goal is to cover as many points as possible. Therefore, we are looking for best z-coordinate for this purpose. Repeat this procedure for every camera in the sorted list. Report how many points are covered, and what is the total displacement for each of them. At end, choose camera that have covered most of points with least best displacement and has least overlap.

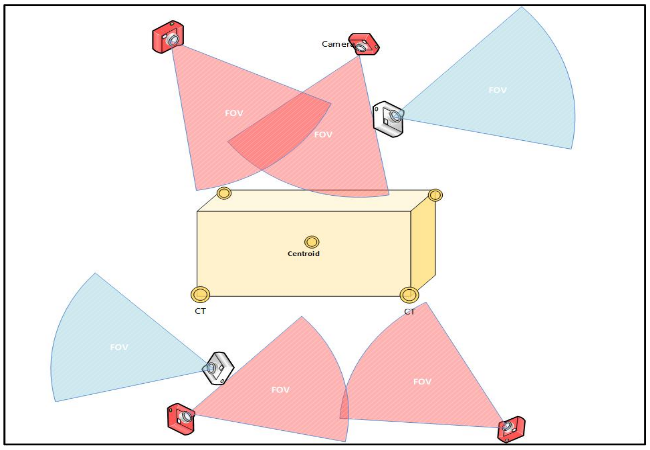

Figure 5.1. Four cameras, two above and two below the object, are actuated by acoustics sensors (Sample case)

To better understanding the idea , check Fig. 5.1 . In the diagram, the topology is showing Four actuated cameras(i.e., Red highlighted Four cameras) because they found at least one of the Five critical points (i.e., P_i X-axis and Y-axis) of object inside their frustum (i.e., All the three camera checking conditions are met). That means to save computing all list of cameras k in the whole topology, we let only those cameras to scan the object. For that reason, we end up with Four cameras eligible to scan the prim. Scanning the prim is not gurantee that cameras are always be choosen. On other words, after each of camera in the selected list of cameras checked the prim, we do the filterization based on distance from initial z poistion and best displacement, coverage rate, and the overlapping rate. In Fig.5.2, we can see that the two cameras are standing to their best displacement in which they cover most of target's critical points. This step, is the final step after a set of filterization of cameras in the list. at the end, these cameras are only cameras are going to adjust their depth, others will be dismissed.

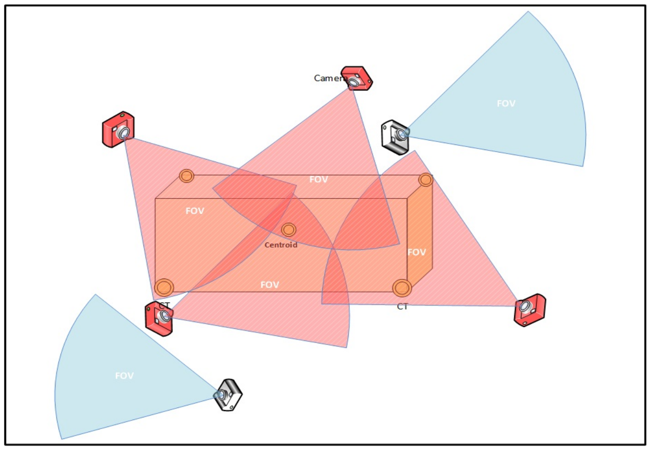

Figure 5.2. Four cameras standing in their best displacement and it shows they cover all of the object

Algorithm 4 Coverage Calculation of Camera Upper the Object

for all $CTotal\ C_i$ upper Object **do**

  Sort $CTotal$ in increasing order to vertical dis. to prism

  if $CameraFrustum < ObjMaxZ$ **then**

    $CameraZ == ObjMin_Z$

    while $CameraZ\ ! > ObjMin_Z$ **do**

      for each Object P_i Five point Do **do**

        if P_i Inside Cam Frustum **then**

          Report P_i as covered,exit

        else

          check Next P_i

        end if

      end for

      Decrease value of $CameraZ$ by certain value

    end while

    Report Total Result of Covered P_{Total} and all correspond $CameraZ$

  end if

end for

Algorithm 5 Coverage Calculation of Camera Below the Object

for all $CTotal$ C_i thatBelow Object **do**
  Sort $CTotal$ in decreasing order to vertical dis. to prism
  if $CameraFrustum > ObjMinZ$ **then**
    $CameraZ == ObjeMin_Z$
    while $CameraZ \; ! < ObjMaxZ$ **do**
      for each Object Pi Five point **do**
        if Pi InsideCam Frustum **then**
          Report Pi as covered and check next
        else
          check Next Pi
        end if
      end for
      Decrease value of $CameraZ$ by certain value
    end while
    Report Total Result of Covered P_{Total} and all correspond $CameraZ$
  end if
end for

CHAPTER 6

PERFORMANCE EVALUATION

This chapter presents the experiment used for finding an efficient camera selection for maximized target coverage. The coverage range for the ultrasonic sensors was 40m. To evaluate the performance of this study's algorithms, Java was used to simulate the approach.

6.1 EXPERIMENT SETTINGS

The simulations were evaluated in a three-dimensional rectangular monitoring area (1000m by 1000m) with the depth adjusted to be 500m. Different random CDS connected topologies were used. We used some of values from table 3.2 with some changes for our study like $d2$ is set in most of cases till 10m. However, we used d2 as 15m for science sake and because the table showed only commercial models and some of cases models of these Factories can be customized for different purposes. The topologies in the experiment consisted of pre-assigned nodes (i.e., acoustic sensors and cameras) in which cameras could comprise 1/4 or 1/5 of the total number of nodes. Instead of using randomly assigned cameras in the topology, camera locations were clustered together. That meant that every fourth node in the topology was a camera. This mechanism was repeated until all cameras were assigned.

To evaluate the performance of this approach, different metrics were set to measure full coverage with the least overlapping among actuated cameras. At the same time, cameras could only move vertically to minimize their moving cost and meet best coverage. The list of metrics evaluated in the experiment is as follows:

- *Coverage*: This metric indicates how many cameras can fully cover the five critical points that represent the target, where P_i denotes one of the total five critical points that represent the target. In other words, the total number of P_is marked as covered represent the coverage percentage of the target. Thus, the formula for coverage percentage can be stated as follows:

$$CoveragePercentage = \frac{P_{covered}}{P_{max}} \tag{6.1}$$

where $P_{covered}$ is total critical points covered by actuated cameras.

- *Cost*: This metric indicates how many of the total number of actuated cameras are used to meet the maximum coverage, where C_{Active} denotes the total number of cameras actuated to cover target. Thus, the formula of cost percentage can be stated as follows:

$$CostPercentage = \frac{C_{Active}}{C_{total}} \tag{6.2}$$

Where C_{total} is the total number cameras in the topology.

The goal of this study was to find a set of cameras that could fully cover the target with the least amount of cost. It is clear that cost and coverage not only involve $P_{covered}$ or C_{Active} but also include a number of detected sensors that participate to sense the passing object, exchange messages with neighboring cameras, and select which cameras should be actuated.

- *Movement Distance*: This metric indicates the total distance for a camera to be traveled vertically to reach its best displacement. That means that after scanning the target for critical points, the camera will decide which camera view frustum has maximum coverage of the target. Cost measurement is calculated as the distance from initial position of camera (i.e., assigned position when the whole topology is deployed) and the position that has the most object coverage. Having the camera with the least distance move while continuing to maintain best coverage were one of the main goals of this study.

Camera setting and some related information (i.e., topology and object setting) form the most important parameters in this stage of the experiment. They are explained as follows:

1- Camera FOV = (69.2V, 92H) and DOF, or in practice near and far planes, was (0.15m - 15m) during the day or (0.15m - 5m) at night. FOV and Frustum view size had more chances to have a high number of cameras cover the object, but more overlapping was also expected.

2- Camera Orientation was one of the most important factors affecting coverage, which is explained in greater detail later on. Camera orientation, or n, = (1, 2, 3), (1, 0, 1), (1, 0, 0) or negative values of each of them to direct a camera to the opposite side of each axis.

3- Topology (Acoustic Sensor and Camera) were set as (750:250) or (800:200). Cameras numbers could be filtered as 1/4, 1/5, or 1/2 of the 1000 total topology nodes; others were acoustic sensors. More cameras mean more possible chances of full coverage, but as with cost, there is a greater chance of overlap, which leads to consuming more time on processing.

4- Object Length Sizes were (80W, 10H, 10L) for this study or (40W, 5H, 5L) in some settings. Of course, the bigger the target is, the more cameras are needed to be involved for coverage. In all testing cases, the velocity of an object was assumed fixed, and once listed sensors recognized the location of the target, that location was considered the target.

These parameters were used in four different scenarios, $Scen1$-$Scen4$, which showed different outputs in the experiment. In all testing cases, 10 different CDS topologies were run 10 times for each setting and some in cases up to 100 times to obtain more robust outputs. The four scenarios were set as follows:

- $Scen1$: This scenario consists of a normal case of checking coverage and has the following parameters: Camera FOV = (69.2V, 92H), DOF = (0.15m - 15m), Camera orientation n = (1, 2, 3), Topology = (750:250), and Object = (80W, 10H, 10L).

- *Scen*2: This case shows that camera orientation is facing width (i.e. X-axis) only and ignores y and Z-axes. Camera FOV = (69.2V,92H), DOF = (0.15m - 15m), Camera orientation n = (1, 0, 0), Topology = (750:250), and Object = (80W, 10H, 10L).

- *Scen*3: In this case, camera number is very low. Camera FOV = (69.2V, 92H), DOF = (0.15m - 15m), Camera orientation n = (1,2,3), Topology = (800:200), and Object = (80W, 10H, 10L).

- *Scen*4: This is a special case in which the environment is darker. Camera FOV = (69.2V, 92H), DOF = (0.15m - 5m) , Camera orientation n = (1, 2, 3) , Topology = (800:200), and Object = (40W, 5H, 5L).

6.2 RESULTS

This section shows the performance results. Four different settings, *Set*1-*Set*4, were applied, and 10 different CDS topologies were run 100 times in the simulation to show different impacts on the algorithms. Testing the simulations produced remarkable outcome in terms of maximum coverage in each of these settings. The simulation results are reported below.

6.2.1 Coverage Rate of Target

The simulation in *Set*1-*Set*4 showed the results given in Figure 6.1. In this diagram, average coverage rate varies from one topology to another. In *Scen*1, 40% of the 10 topologies fully covered the object every time. There was only one topology, Top9, that covered 20% (i.e. covered only 1 critical P of the target's total critical P_t) whereas the rest of the topologies successfully covered a range of 60%-80% of the total target. In brief,

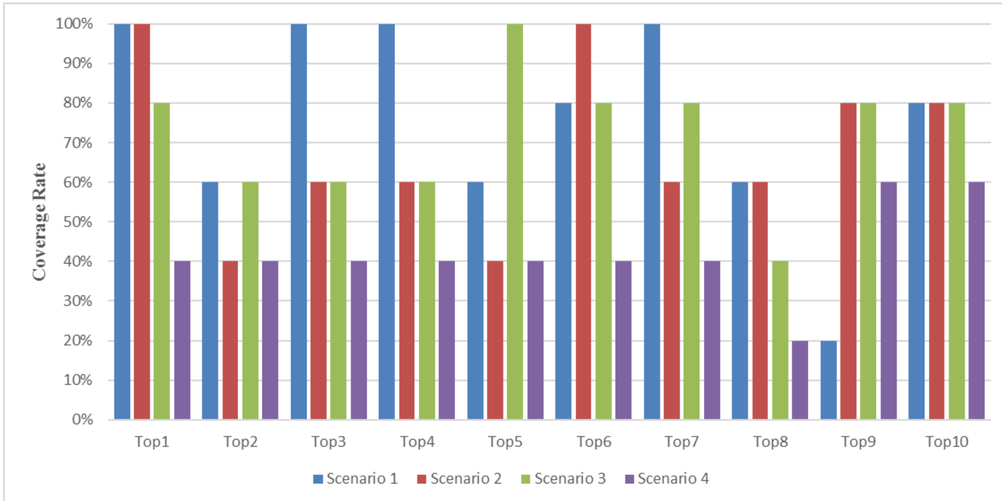

Figure 6.1. Different Setting that show percentage of coverage per each CDS topology.

in coverage point of view, this was a very good output and suggested that $Scen1$ should be used in normal status.

In $Scen2$, however, when orientation was set to face only the X-axis, which can be parallel with the width of the target, or when the number of total cameras in the topology was reduced, as in $Scen3$, the majority of topologies showed similarities in both scenarios. However, in $Scen4$, camera DOF was very low, and as a result the object was reduced to half its original size. This part was tested more than 100 times to show full coverage. In fact, as the graph shows, only 20% of the topologies could cover up to 60% of the total target in best case scenarios. The bottom line in this section was that three scenarios were able to cover 100% of the object most of the time, and if not, the topologies could cover more than 60% of the target, while no topology in any of the four scenarios showed zero coverage.

Another high priority for coverage was avoiding overlaps among activated cameras. Different scenario settings were tested in 10 different topologies and the results were impressing. In Figure 6.1, the overlapping cases occurred for most of the topologies relying on between 15% to 30+% of the total target. One case showed overlapping rise to

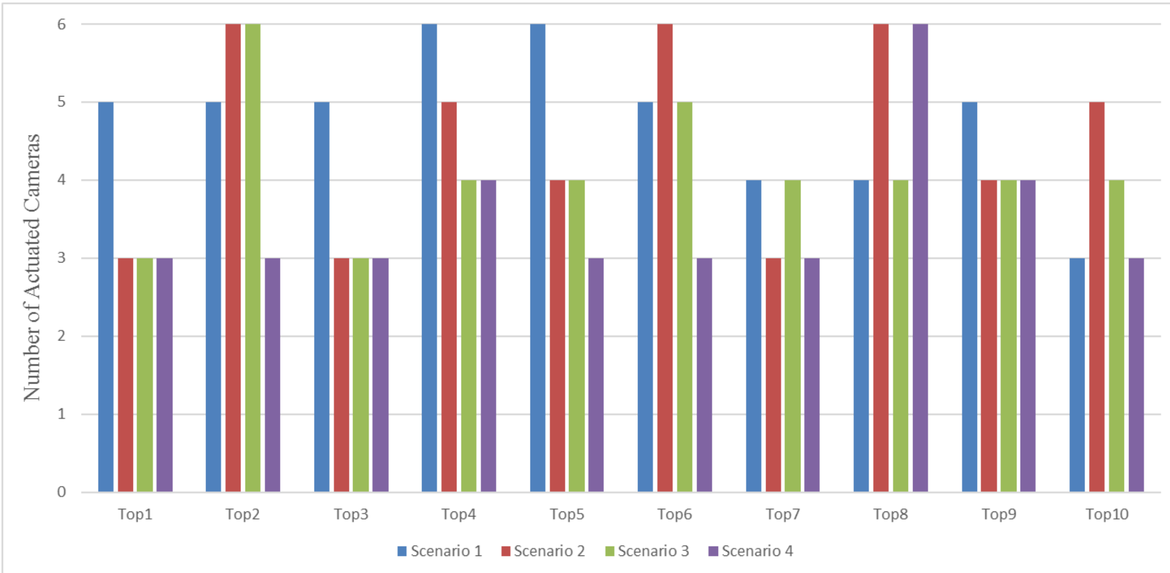

Figure 6.2. Total Number of Activated Cameras Participating in each Scenario.

60% of total target coverage. That was reasonable for $Scen1$ since it showed a high coverage rate in most cases. As mentioned before, cases like $Scen3$ and $Scen4$ were expected to result in low rates of overlapping with few participating cameras in most cases. However, $Scen2$ showed a surprising result. The overlapping rate remained steady at a low rate of less than 30% overlapping. This led this scenario to compete with $Scen1$, which was the best solution for this study's approach. One possible reason for this was that the cameras were parallel with the width of the object, which gave more opportunities for the algorithms to filter any expected overlapping among cameras that rely on the two width borders of the object (i.e. 80 W, 10 H, 10 L).

In Figure 6.2, the graph shows different cameras participating in each scenario. When the goal is to save energy for cameras, having fewer participating cameras is always preferable if and only if maximum coverage is met. Obviously, this could lead to situations requiring compromise in different scenarios. Since an object has five critical points, the best case scenario should be to have five or fewer cameras. That means that each point is covered by at maximum one camera, and the best case scenario is finding

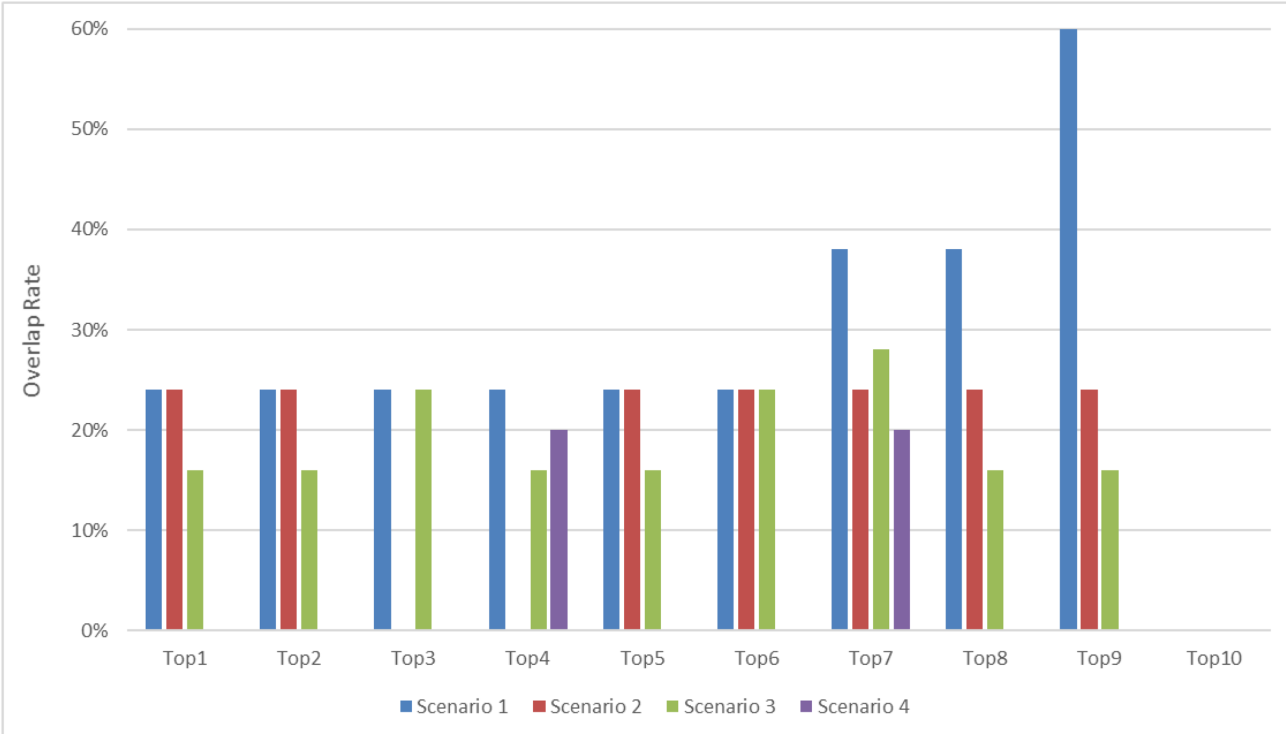

Figure 6.3. Percentage of Overlapping in Different Settings.

one camera can cover most of the five points. Numbers in the graph show after full coverage has occurred in each scenario. For example, in $Scen4$, participating cameras remained stable in most topologies for three cameras. One reason for that could be because in that scenario there was no full coverage in any of the topologies, even the overlapping situation in Top8 that will be explained shortly. On the other hand, $Scen1$ and $Scen2$ showed the most of cases of many cameras participating in full coverage. That was to be expected since these two scenarios have the highest coverage rates, while $Scen3$ showed interesting results. Since there was a small amount of C_{total} in $Scen3$, that directly affected the number of activated cameras in different topologies.

Avoiding overlaps among activated cameras is another high prioriy in our approch. We tested different scenarios settings in 10 different topologies and the results were impressed. In Fig.6.3, the overlapping cases occure for most of topologies set on between

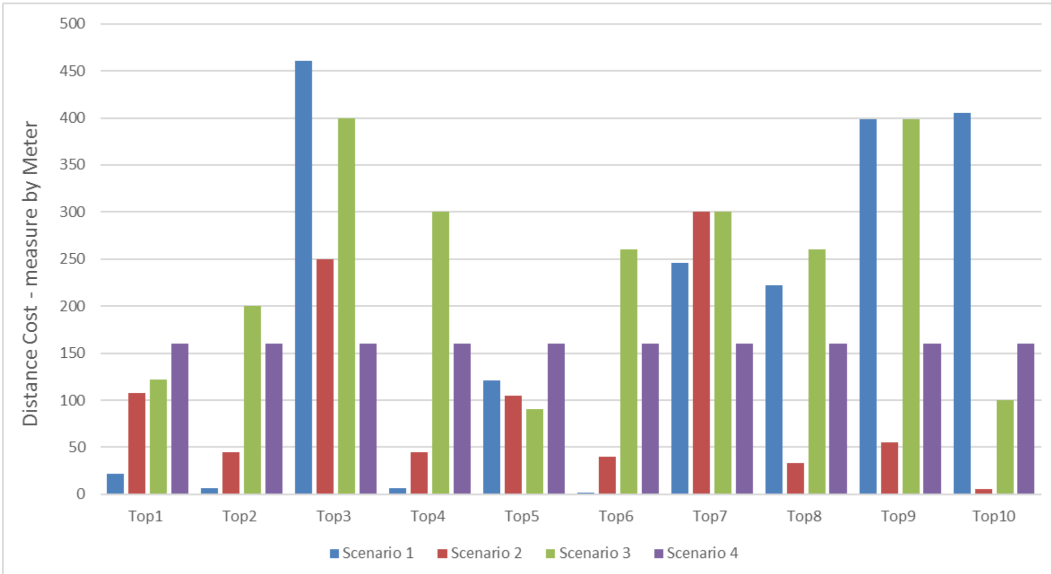

Figure 6.4. Different Camera Costs shown in multi-topologies

15% to mid-of 30s% of total Target. One case shows overlap rocketed 60% of total target coverage. That can be reasonable for $Scen1$ since it showed high level of coverage rate in most of cases. As we mentioned before, cases like Scen3 and Scen4 are always expected to get low rate of overlap with not much of particpated cameras on most of cases. However, $Scen2$ has shown surprised result. the overlapping rate remained steady at low rate of less than 30% overlapping rate. This will lead this scenario to compete Scen1 of which is best to be used as best solution of our approch. One possible reason is that because cameras are in parallel with width of object(i.e., width of object is 80m whereas height and length are 10m each. That gives more chance for algorithms to filter any expected overlapping among cameras that set on between the two width borders of object.

6.2.2 Cost of Target Coverage

When talking about cost of coverage, not only did this involve number of participating cameras, but also how much activated cameras consumed while adjusting

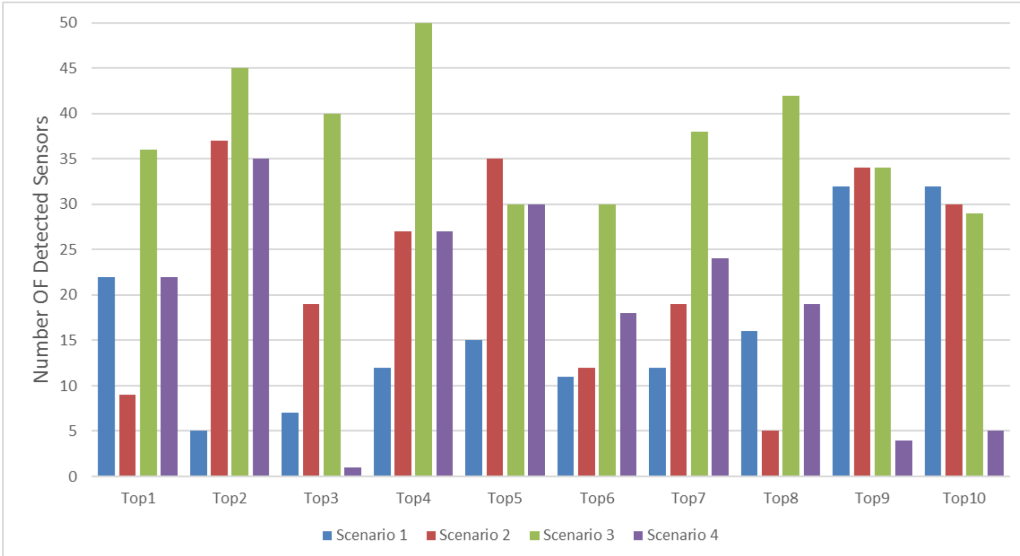

Figure 6.5. Detected Acoustic Sensors in All Topologies

their depth to reach their best displacement. For example, Figure 6.4 shows all scenarios and their impact on camera cost in the topologies. $Scen2$ appeared the best in terms of cost, whereas $Scen1$ displayed the worst and best case scenarios. The cameras being randomly distributed throughout the monitoring area, the cost to move up or down consumed more energy in some cases, while in others, there was no need to move, and the cost was 0. The greatest cost occurred when a camera had to move all the way from the top of the area to the bottom. This was due to the large number of participating cameras in the coverage area, which allowed extra cameras to be activated even with smart filtering algorithms under this approach. Figure 6.5 shows detected sensors in different scenarios.

In brief, coverage and cost are always in compromis scale. The algorithms used worked very well to meet the goals of this study.

6.3 DISCUSSION

Assigned the results showed significantly better coverage with this approach. This approach resulted in coverage in a number of scenarios, showing that even in the worst case scenarios; maximum coverage was still obtained with little cost from overlap and camera adjustment. It is obvious that these results are not gurantee full coverage of such target for all cases but it did gurantee to choose only best cameras that cost less energy and least overlapping in all scenarios. For example, $Scen4$ showed the worst scenario in our experiment. In fact, even if there is a sacrification in movement of cameras , the result still showed reasonable coverage of the target. In brief, our experiment showed that there is always coverage in either best cases or worst cases scenarios.

CHAPTER 7

CONCLUSION

The distributed approach is currently very demanding in UWASNs due to the nature of battery-powered acoustic and multimedia sensors. Monitoring an area in a 3-D environment is challenging because of several important aspects that should be taken into consideration.

In this thesis, we presented an algorithm for maximizing the camera coverage of an object that is detected by acoustics sensors in underwater. Our proposed algorithm determined the closest cameras to move such that the target coverage will be maximized. Since the movements are in vertical directions, the approach does not guarantee full coverage. Nonetheless, it minimizes the movement of cameras. The approach successful is showing how only the best-positioned cameras can be activated to cover an object.

Future studies could focus on assuring full coverage even under the worst case scenario in a topology, namely in a dark or low-light environment. Another avenue of research could be fault-tolerance of messages among topologies and looking for using UAVS to participate in obtaining full coverage.

REFERENCES

[1] E. Sozer, M. Stojanovic, and J. Proakis. Underwater acoustic networks. In *IEEE Journal of Oceanic Engineering*, volume 25, pages 72–83, Jan. 2000.

[2] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li. Research challenges and applications for underwater sensor networking. In *IEEE Wireless Communications and Networking Conference (WCNC'06)*, Las Vegas, Nevada, April 2006.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: Research challenges. In *Ad Hoc Networks*, volume 3, pages 257–279, May 2005.

[4] Jun-Hong Cui, Jiejun Kong, Mario Gerla, and Shengli Zhou. Challenges: building scalable and distributed underwater wireless sensor networks (uwsns) for aquatic applications. *Channels*, pages 1–17.

[5] F. Senel, K. Akkaya, and T. Yilmaz. Autonomous deployment of sensors for maximized coverage and guaranteed connectivity in underwater acoustic sensor networks. In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pages 211–218, Oct 2013.

[6] Ultrasonic Sensor for Underwater. http://www.alibaba.com/product-detail/2013-underwater-ultrasonic-sensor$_9$30548088*.html*.

[7] V. Ravelomanana. Extremal properties of three-dimensional sensor networks with applications. In *IEEE Transactions on Mobile Computing*, volume 3, pages 246–257, July/September 2004.

[8] S. N. Alam and Z. Haas. Coverage and connectivity in three-dimensional networks. In *Proc. of ACM Int'l Conf. on Mobile Computing and Networking (MobiCom'06)*, Los Angeles, CA, Sept. 2006.

[9] D. Pompili, T. Melodia, and I. F. Akyildiz. Deployment analysis in underwater acoustic wireless sensor networks. In *Proc. of ACM Int'l Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[10] Kemal Akkaya and Andrew Newell. Self-deployment of sensors for maximized coverage in underwater acoustic sensor networks. *Comput. Commun.*, 32(7-10), May 2009.

[11] Erdal Cayirci, Hakan Tezcan, Yasar Dogan, and Vedat Coskun. Wireless sensor networks for underwater survelliance systems. *Ad Hoc Networks*, 4(4):431 – 446, 2006.

[12] M. Erol, F.L.M. Vieira, and M. Gerla. Auv-aided localization for underwater sensor networks. In *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pages 44–54, Aug 2007.

[13] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 154–165, New York, NY, USA, 2005. ACM.

[14] V. Davies T. Camp, J. Boleng. 'a survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[15] J. Cui, Z. Zhou, and A. Bagtzoglou. Scalable localization with mobility prediction for underwater sensor networks. In *Second workshop on Underwater networks (WuWNet)*, Montreal, Quebec, Canada, 2007.

[16] Diba Mirza and Curt Schurgers. Motion-aware self-localization for underwater networks. In *Proceedings of the Third ACM International Workshop on Underwater Networks*, WuWNeT '08, pages 51–58, New York, NY, USA, 2008. ACM.

[17] A. Caruso, F. Paparella, L. Vieira, M. Erol, and Gerla M. The meandering current mobility model and its impact on underwater mobile sensor networks. In *27th Int'l*

Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'08), Phoenix, Arizona, April 2008.

[18] M. Erol, L. Vieira, A. Caruso, F. Paparella, M. Gerla, and Oktug S. Multi stage underwater sensor localization using mobile beacons. In *The Second International Workshop on Under Water Sensors and Systems (in conjunction with SENSORCOMM'08)*, Cap Esterel, France, August 2008.

[19] T. Pribanic, M. Cifrek, and S. Tonkovic. The choice of camera setup in 3d motion reconstruction systems. In *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, volume 1, pages 163–165 vol.1, 2000.

[20] Thomas D. C. Little, Janusz Konrad, and Prakash Ishwar. A wireless video sensor network for autonomous coastal sensing, 2007.

[21] FOV-Calculation. http://www.panohelp.com/lensfov.html. Accessed: 2014-10-2.

[22] view-frustum-culling camera frustum. http://www.lighthouse3d.com/tutorials/view-frustum-culling/geometric-approach-extracting-the-planes/. Accessed: 2010-09-30.

[23] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, Oct 2001.

[24] S. Colonnese, F. Cuomo, and T. Melodia. An empirical model of multiview video coding efficiency for wireless multimedia sensor networks. *Multimedia, IEEE Transactions on*, 15(8):1800–1814, Dec 2013.

[25] S. Pudlewski, A. Prasanna, and T. Melodia. Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. *Mobile Computing, IEEE Transactions on*, 11(6):1060–1072, June 2012.

[26] S. Soro and W.B. Heinzelman. On the coverage problem in video-based wireless sensor networks. In *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*, pages 932–939 Vol. 2, Oct 2005.

[27] Liang Liu, Huadong Ma, and Xi Zhang. On directional k-coverage analysis of randomly deployed camera sensor networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2707–2711, May 2008.

[28] Manish Kushwaha and Xenofon Koutsoukos. Collaborative 3d target tracking in distributed smart camera networks for wide-area surveillance. *Journal of Sensor Actuator Networks*, 2(2):316 – 353, 2013.

[29] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 115–121, New York, NY, USA, 2003. ACM.

[30] Yi Hong, Donghyun Kim, Deying Li, Biaofei Xu, Wenping Chen, and A.O. Tokuta. Maximum lifetime effective-sensing partial target-coverage in camera sensor networks. In *Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on*, pages 619–626, May 2013.

[31] N. Brahim, D. Gueriot, S. Daniel, and B. Solaiman. 3d reconstruction of underwater scenes using image sequences from acoustic camera. In *OCEANS 2010 IEEE - Sydney*, pages 1–8, May 2010.

[32] Umberto Castellani, Andrea Fusiello, and Vittorio Murino. Underwater environment modelling by fast 3d mosaicing.

[33] Lijun Xie and Xi Zhang. 3d clustering-based camera wireless sensor networks for maximizing lifespan with minimum coverage rate constraint. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 298–303, Dec 2013.

[34] Han Chong, Sun Lijuan, Xiao Fu, Guo Jian, and Wang Ruchuan. Fovs correlation model for wireless multimedia sensor networks based on three-dimensional perception. *Advances in Information Sciences Service Sciences*, 4(3):27, 2012.

[35] Han Chong, Sun Lijuan, Xiao Fu, Guo Jian, and Wang Ruchuan. A camera nodes correlation model based on 3d sensing in wireless multimedia sensor networks. *International Journal of Distributed Sensor Networks*, pages 1 – 9, 2012.

[36] Huadong Ma and Yonghe Liu. Correlation based video processing in video sensor networks. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 2, pages 987–992 vol.2, June 2005.

[37] K. Muljowidodo, M. A. Rasyid, N. SaptoAdi, and A. Budiyono. Vision based distance measurement system using single laser pointer design for underwater vehicle. *INDIAN JOURNAL OF MARINE SCIENCES*, 38(3):324 – 331, 2009.

[38] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh. Cmucam3: an open programmable embedded vision sensor. In *IEEE Transactions on Mobile Computing*, volume RI, pages 07–13, July/September 2007.

[39] M. Alnuaimi, F. Sallabi, and K. Shuaib. A survey of wireless multimedia sensor networks challenges and solutions. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 191–196, April 2011.

[40] Underwater Camera. http://www.amazon.com/color-fishing-underwater-camera-bright/dp/b009359qeo/ref=sr$_{13}$?$ie = utf8qid = 1395265744sr = 8 - 3keywords = underwater + ccd + video + camera.$

[41] Underwater Camera. http://chinatopwell.en.alibaba.com/product/397979962-210130288/360$_degree_rotatable_underwater_fishing_camera.html.$

[42] Underwater Camera. http://granfoo.en.alibaba.com/product/356028534-
$213289761/2014_{h}ot_{s}elling_{r}emote_{c}ontrol_{5}00m_{s}ony_{c}cd_{7}00tvl_{3}60_{d}egree_{c}ctv_{b}ore_{d}ownhole_{i}nspection_{u}nderwat$

[43] Y. Maki, S. Kagami, and K. Hashimoto. Accelerometer detection in a camera view based on feature point tracking. In *System Integration (SII), 2010 IEEE/SICE International Symposium on*, pages 448–453, Dec 2010.

[44] Reza Oji. An automatic algorithm for object recognition and detection based on ASIFT keypoints. *CoRR*, abs/1211.5829, 2012.

[45] Jean-Michel Morel and Guoshen Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469, April 2009.

[46] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme. Sensor network configuration and the curse of dimensionality. In *3rd IEEE Workshop on Embedded Networked Sensors*, Cambridge, MA, May 2006.

[47] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. In *Elsevier Ad Hoc Network Journal*, volume 6, pages 621–655, 2008.

[48] G. Wang, G. Cao, and T. La Porta. Movement-assisted sensor deployment. In *23rd Int'l Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04)*, Hong Kong, March 2004.

[49] G. Wang, G. Cao, and T. La Porta. Proxy-based sensor deployment for mobile sensor networks. In *1st IEEE Int'l Conference on Mobile Ad-hoc and Sensor Systems (MASS'04)*, Fort Lauderdale, FL, October 2004.

[50] G. Wang, G. Cao, T. La Porta, and W. Zhang. Sensor relocation in mobile sensor networks. In *24th Int'l Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05)*, Miami, FL, March 2005.

[51] J. Wu and S. Yang. Smart: A scan-based movement assisted sensor deployment method in wireless sensor networks. In *24th Int'l Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05)*, Miami, FL, March 2005.

[52] M. Gerla J.-H. Cui, J. Kong and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. In *Special Issue of IEEE Network on Wireless Sensor Networking*, May 2006.

[53] J. Partan, J. Kurose, and B. N. Levine. A survey of practical issues in underwater networks. In *Proc. of ACM Int'l Workshop on UnderWater Networks (WUWNet'06)*, Los Angeles, CA, Sept. 2006.

[54] V. Chandrasekhar, W. Khoon, G. Seah, Y. S. Choo, and H. V. Ee. Localization in underwater sensor networks: survey and challenges. In *Proc. of ACM Int'l Workshop on UnderWater Networks (WUWNet'06)*, Los Angeles, CA, Sept. 2006.

[55] V. Bokser, C. Oberg, G. S. Sukhatme, and A. A. Requicha. A small submarine robot for experiments in underwater sensor networks. In *Int'l Federation of Automatic Control Symposium on Intelligent Autonomous Vehicles*, 2004.

[56] Carrick Detweiler, Marek Doniec, Iuliu Vasilescu, Elizabeth Basha, and Daniela Rus. Autonomous depth adjustment for underwater sensor networks. In *Proc. of ACM Int'l Workshop on UnderWater Networks*, WUWNet '10, New York, NY, USA, 2010.

[57] Christian Bettstetter. On the connectivity of ad hoc networks. *The computer journal*, 47(4):432–447, 2004.

[58] Jie Wu and Hailan Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of the 3rd Int'l workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, New York, NY, USA, 1999.

[59] K. Akkaya and S. Janapala. Maximizing connected coverage via controlled actor relocation in wireless sensor and actor networks. In *Elsevier Computer Networks*, volume 52, pages 2779–2796, 2008.

[60] Navneet Malpani, Jennifer L. Welch, and Nitin Vaidya. Leader election algorithms for mobile ad hoc networks. In *Proc. of the 4th Int'l workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '00, New York, NY, USA, 2000.

[61] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[62] A. Vulpiani M. Cencini, G. Lacorata and E. Zambianchi. Mixing in a meandering jet: a markovian approximation. *J. Phys. Ocean.*, 29(10):25782594, 1999.

[63] A. S. Bower. A simple kinematic mechanism for mixing fluid parcels across a meandering jet. *J. Phys. Ocean.*, 21(1):173–180, 1991.

[64] R. M. Samelson. Fluid exchange across a meandering jet. *J. Phys. Ocean.*, 22:431–444, 1992.

[65] A. J. Chorin and O. H. Hald. Stochastic tools in mathematics and science. *Surveys and Tutorials in the Applied Mathematical Sciences*, 2006.

[66] A. Provenzale C. Pasquero and A. Babiano. Parameterization of dispersion in two-dimensional turbulence. *J. Fluid Mech.*, 439:279–303, 2001.

[67] Henrik Aans, Anders Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18 – 35, 2012.

[68] Wireless underwater camera sensor camera model. http://www.lighthouse3d.com/tutorials/view-frustum-culling/geometric-approach-

extracting-the-planes/. Accessed:
2010-09-30.

VITA

Graduate School
Southern Illinois University

Abdullah AlBuali

abdullah.albuali@gmail.com

King Faisal University, Hofuf, Saudi Arabia
Bachelor of Science, Computer Science, May 2009

Thesis Title:
Efficient Camera Selection For Maximized Target Coverage In Underwater Acoustic Sensor Networks

Major Professor: Dr. Kemal Akkaya

Publications:

[C1] A. alGhamdi, M. Alghamdi, **A. Albuali** and H. Hexmoor: An Approach for Computing Social Capital, International Journal of Information Systems and Computer Sciences, 2(6), November, 2013