MAXIMIZING SENSOR COVERAGE USING DISPLACEMENT TO REMOVE OVERLAPS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Kareemullah Khan Fazal

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

IMPROVING SENSOR COVERAGE USING

DISPLACEMENTS TO REMOVE OVERLAPS

By

KAREEMULLAH KHAN FAZAL

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Fazal, Kareemullah Khan, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, October 2011. Maximizing Sensor Coverage Using Displacement to Remove Overlaps. Major Professor: Dr. Kendall Nygard.

Sensor networks are wireless networks with small, low-cost sensors which collect and disseminate environmental data. They are used for monitoring and controlling physical environments from remote locations with better accuracy. Sensors have a lot of constraints and challenges. One of the main concerns is the coverage and redistribution of the sensors, without any overlap, in order to achieve maximum coverage with limited usage of energy.

The first section of the paper gives a brief overview of the sensors, the sensor network, the various applications where sensors are used, and challenges faced in setting up a sensor network. The next section outlines the problems faced in sensor coverage, followed by a description of the random displacement of the mobile sensors during overlap technique. The following section details the application used to generate the prototype of the sensor network. This section is followed by a visualization section where images of the sensor network are described in detail. The subsequent section provides details about the results of the test cases. The last section of the paper lists the assumptions and constraints that were kept in mind while developing the application, finishing with the conclusion and references.

In the prototype of the sensor network, the sensors will be deployed in a given geographical area. The sensor nodes being deployed will be static as well as mobile sensors. The assumption made for this project is that the environment in which the sensors

are going to deployed is uncontrolled. The deployed sensors will be searched using a particular procedure to find the overlapping mobile and static sensors. The random displacement technique is used to move the mobile sensor when there is an overlap between two sensors in order to avoid overlapping between the two sensors to enhance the quality of service and to increase the coverage in a given area.

ACKNOWLEDGEMENTS

I would like to thank Dr. Kendall E. Nygard for his continued support, help, and guidance. I would like to thank my family and friends who supported and motivated me to accomplish this task, especially my wife, Aaliyah Fathima Masa-Kareem. Without her this work would not have been possible.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

1. INTRODUCTION 1

2. LITERATURE REVIEW 3

3. PROBLEM DESCRIPTION IN SENSOR COVERAGE 8

4. RANDOM DISPLACEMENT OF MOBILE SENSORS DURING OVERLAP 13

5. HIGH-LEVEL DESIGN 16

5.1. Class Diagram 16

5.2. Sequence Diagram 18

5.2.1. Deploy Sequence Diagram 18

5.2.2. Move Sequence Diagram 19

5.2.3. Move-All Sequence Diagram 22

6. APPLICATION 24

6.1. Inputs Provided 24

6.2. Color Coding 25

6.3. The Task Bar 25

6.3.1. 'Data' Tab 25

6.3.2. 'Deploy Sensor' Tab 25

6.3.3. 'Move' Tab 26

6.3.4. 'Move All' Tab 26
6.4. The Combo Box 26
6.4.1. 'Total Area' Field 26
6.4.2. 'Coverage Area' Field 27
6.4.3. 'Non-Coverage Area' Field 27
6.4.4. 'Overlap Area' Field 27
6.4.5. 'Gain' Field 28
6.4.6. 'Overview' Field 28
7. VISUALIZATION 29
8. RESULTS 36
8.1. Test Case 1 36
8.2. Test Case 2 37
8.3. Test Case 3 37
8.4. Test Case 4 42
9. CONSTRAINTS 44
10. CONCLUSION 45
REFERENCES 46

LIST OF TABLES

Table Page

1. Test Case 1 38

2. Test Case 2 39

3. Test Case 3 40

4. Test Case 4 43

LIST OF FIGURES

Figure | Page

1. Grid Diagram. 14

2. Class Diagram. 17

3. Deploy Sequence Diagram. 20

4. Move Sequence Diagram. 21

5. Move - All Sequence Diagram. 23

6. Setting Form for Data Entry. 29

7. Sensor Network with Overlaps. 31

8. Sensor Network Without Overlaps. 32

9. Data Entry with 25% Overlap Threshold. 33

10. Sensor Network with Overlap > 25%. 34

11. Relocation of Overlapping Sensors > 25%. 35

12. Comparison of Gain Data from Test Cases 1, 2 and 3. 41

13. Gain Coverage with Incremental Mobile Sensors. 42

1. INTRODUCTION

Sensor networks are one of the latest technologies where numerous research activities are taking place recently. Advancement in this area was made possible due to wireless communication and micro-sensing technologies. The sensors are usually deployed in a small-scale or large-scale network. Sensors are used in many applications, but if you consider large-scale networks, then they are mostly used for military applications, environmental observations, etc.

Although sensor networks can be used in numerous applications, they have some limitations. One of the challenges with sensor networks is power consumption (Liu and Towsley), and the other problem is the sensor coverage (Meguerdichian et al.). On large-scale networks, the area is usually uncontrolled, and deployment of the sensors is usually achieved through randomly scattering the sensors. When the sensors are scattered randomly there are very high chances of network coverage overlap, and as a result, there will be less coverage in the given sensor area.

This project focuses mainly on achieving maximum sensor coverage by randomly deploying the sensor when there is an overlap. We deploy the static and mobile sensors in an area specified by the user. When there is any overlap between a static and a mobile sensor or between two mobile sensors, there is duplication of data that are being transmitted, and also the quality of service is reduced drastically. Better quality can be achieved by moving the mobile sensors to an empty grid cell using the random displacement technique. Based on the coverage, the mobile sensors with maximum overlap move to a grid cell that needs maximum coverage (Meguerdichian et al.). This process will be followed by the second mobile sensor with maximum overlap, and it is continued until we achieve no overlap between the sensors.

2. LITERATURE REVIEW

The field of sensor networks has flourished with ongoing research activity due to the vast, long-term economic potential that the field possesses. Sensors are small, low-powered devices with the capability to provide wireless communication over short distances. A sensor has the capability to perceive the environment where it is established or the phenomenon that justified its implementation. It must also be able to transmit the perceived data. Such sensors are called scalar sensors. There exist other types called multimedia sensors. They include equipment that is able to perceive and broadcast audio streams, video streams, or fixed images. The sensors usually have limited resources. A sensor node contains a transmitter/receiver with a power unit, a sensing unit, and a storage unit. Sensor nodes have been used across numerous applications. Although older sensor nodes had limitations, the newer sensor nodes are more sophisticated and have the capacity to build large-scale sensor networks. A wireless sensor network is composed of a great number of sensors, deployed massively for the observation of a phenomenon (Tagne et al.). A sensor network usually consists of a number of sensor nodes with computation, storage, and communication capabilities. The sensor network environment may have many sensor nodes, each capable of collecting, storing, and processing environment information, as well as communicating with the neighboring nodes (Du et al.; Katib; Liu and Towsley; Meguerdichian et al.; Poduri and Sukhatme.).

A sensor network deployment can usually be categorized as either a dense deployment or a sparse deployment. A dense deployment has a relatively high number of sensors in the given field of interest while a sparse deployment would have fewer sensors.

The dense deployment model is used in situations where it is very important for every event to be detected or when it is important to have multiple sensors cover an area. Sparse deployments may be used when the cost of the sensors make a dense deployment prohibitive or when you want to achieve maximum coverage using the minimum number of sensors (Mulligan and Ammari).

The coverage of a sensor network can be classified as a static or a dynamic configuration. In a static configuration, every point in the location is under the sensor network shadow. Depending on the environment size, complexity, and range of the nodes, the sensor network should have a certain number of sensor nodes which are critical to achieve static coverage. However, if the environment is an unknown or a large terrain, static configuration is not feasible. In this case, dynamic coverage is used to achieve sensor coverage by covering the terrain with constant motion. Because dynamic coverage need not remain with a particular configuration, it is difficult to achieve. The cost of sensors adds to the overall complexity of a dynamic system.

Sensor networks have the ability to revolutionize the entire concept of data collection and data reporting. Because of their economic potential, there has been tremendous research on this topic. Advances with micro-sensor and communication technology have made it feasible to build sensors with sensing, processing, and wireless communication capabilities. There has also been research on sensor coverage. Many procedures have been devised to minimize the sensor coverage problem and to make sensor networks more efficient and feasible. Most procedures have one thing in common; the sensing ability of the sensor node is directly dependent on the distance (Katib). A sensor network can be formed by deploying specialized sensors to perform sensing tasks in the target environment.

The sensor nodes used to build sensor networks reduce the amount of communication to perform their specialized tasks. They implement sophisticated protocols as well as distributed and/or local computation. Depending on the sensor network's environment, sensor nodes have the capability to switch to power-saving modes (Meguerdichian et al.). Hence, the flexibility of installing and configuring a sensor network has greatly improved.

The sensor nodes can be deployed in a controlled environment or an uncontrolled environment. A controlled environment can be an office, a house, or even a warehouse. An uncontrolled environment can be a large agricultural field or a battlefield. A sensor network can be deployed in a controlled environment with much ease and with fewer challenges compared to deployment in an uncontrolled environment. Sensor nodes can be manually deployed for a sensor network in a controlled environment, whereas it is infeasible to manually deploy the sensor nodes for a sensor network in an uncontrolled environment. The sensor nodes are scattered randomly to the specific target area to form a network or networks. In an uncontrolled environment, there are limitations in deploying sensors randomly, mainly because it is infeasible to visit the large number of sensor nodes and to change their configuration (Camtepe and Yener).

A lot of research activity has recently been dedicated to sensor networks, and due to the advances in sensor-network technology, many applications of sensor networks continue to emerge. Sensor networks can be implemented in different areas, both across existing applications as well as in new applications. Some applications include habitat monitoring, home monitoring, environmental and battlefield surveillance, and hostile or large agricultural area surveillance. Most applications involve the deployment of large-scale sensor networks where a large number of sensors are deployed in a vast topographical region. One of the many new applications for sensor networks is monitoring areas at high risk for forest fires. The use of sensor network can significantly reduce the forest-fire detection time, thus resulting in early extinguishing efforts. The cost of deploying the sensor nodes for a sensor network is minimal compared to the cost of losses sustained in the forest fires because the sensor nodes are self-configuring and do not require constant monitoring (Kung). The sensor network can be used as an alternative to several existing technologies that use considerably more resources than the sensor network. As an example, the hard-wired monitoring units for the central air-conditioning unit in a large office building have substantial wiring and maintenance costs. Replacing them with wireless sensor nodes improves the efficiency and the quality of the system. The wireless sensor nodes also allow reconfiguration and customization in the future.

There are various issues and challenges faced by sensor networks. Some problems are the location, deployment, and optimization issues. The most basic and fundamental issue in a sensor network is the coverage (Batalin and Sukhatme; Du et al.; Huang and Tseng; Katib; Liu and Towsley; Poduri and Sukhatme). This paper focuses on the coverage issue. The quality of service for a sensor network is mainly determined by the coverage of the sensor network. By determining the coverage of a sensor network, the weak points/areas in the sensor network can be determined. The coverage provides suggestions for the future deployment of additional sensor nodes and, in this manner, improves the overall quality of the sensor network.

The coverage in a sensor network depends on many factors (Fan and Jin). The most important factor is the property of the terrain. The ease of deployment and placement of the sensors is when the terrain is plain with no obstacles rather than when the terrain has obstacles like hills, water, etc. By optimizing the number of sensors and with the efficient placement of the sensors, the sensor network can reduce the problem of sensor coverage, optimizing the sensor network. In the worst-case coverage, the quality of service is measured by finding areas of lower observability from sensor nodes and detecting violated regions. In the best-case coverage, the main concern is finding high observability areas from sensors and identifying the best support regions.

3. PROBLEM DESCRIPTION IN SENSOR COVERAGE

Coverage in a sensor network is usually defined as a measure of how well and how long the sensors are able to observe the physical space. Sensor-network coverage is the measure of the service quality, and it represents the quality of surveillance that the network can provide. It is determined by how well the region of interest is monitored by sensors and how effectively a sensor network can detect intruders. The coverage depends on various network parameters for different application scenarios. Coverage in many sensor-network applications can be viewed from either a best case or worst case point of view (Mulligan and Ammari). The best-case point of view determines the areas of high coverage, and the worst-case point of view determines the areas of lower coverage. Looking at coverage from both views is helpful to solve different problems.

There has been substantial research in the area of sensor-network coverage. One fundamental issue in sensor networks is the coverage problem, which reflects how well a sensor network is monitored or tracked by sensors. In addition to tracking and deployment, coverage is one of the major issues with a sensor network. The better the coverage, the better the sensor network's performance is in a given region. Hence, the coverage is directly proportional to the performance of a sensor network (Li and Yu; Pottie).

A sensor network consists of a number of sensors which are deployed in the region of interest. In many applications, static sensors are used; they remain stationary after their initial deployment. By using the initial network configuration, the coverage of such a static sensor network is determined; the coverage remains unchanged over time. Recently, the sensors have been deployed on mobile platforms. Sensors that have the ability to relocate after they are deployed are known as mobile sensors. The coverage of a mobile sensor network depends not only on the initial network configurations, but also on the mobility behavior of the sensors. Sensor mobility is exploited to obtain a new stationary configuration that improves coverage after the sensors move to their desired locations (Mulligan and Ammari). Sensor networks with mobile sensors usually start with a random deployment, and sensors relocate to the optimal locations.

The nodes that are selected for a sensor network can be either a homogeneous or heterogeneous group. A homogeneous group is a group in which all nodes have the same capabilities. A heterogeneous group is one in which some nodes are more powerful than others. In this paper, we will be using a homogeneous sensor group for the random displacement of the mobile sensors during overlap technique.

Sensor networks are a rapidly growing area for research and commercial development (Mulligan and Ammari). Sensor networks are used to monitor a given field of interest for changes in the environment. They are very useful for military, environmental, and scientific applications. Mobile sensor networks are used in applications where traditional deployment mechanisms are not suitable, such as in a hostile environment where sensors cannot be manually deployed or air-dropped. Coverage in an uncontrolled environment, like a forest or a battlefield, is more difficult to achieve when compared to coverage in a controlled environment like an office. Because it is infeasible to deploy sensor nodes manually in an uncontrolled environment, the sensors have to be randomly deployed to form the sensor network. In application scenarios such as atmospheric and oceanic environmental monitoring, sensors move with the surrounding air or ocean currents. Randomly scattering the sensors in a sensor network has certain issues such as the high-risk areas might not get covered; there might be an overlap between sensors. The overlap between the sensors will result in reduced performance of the sensor network as more than one sensor will be recording the same data for the region.

The most important factor to consider in the development of a coverage scheme is energy constraints (Mulligan and Ammari). Most research with random deployments for sensors regards the ability to maintain coverage while minimizing the amount of energy expended. Sensor nodes use a battery for their energy source, and in most applications, once the sensors are deployed, battery replacement is not feasible. Therefore, prolonging battery life and conserving energy become very important. There are several methods available. One method is to place unused sensors in a low-energy sleep mode to preserve energy. Another method is to adjust the transmission range so that the sensor nodes only use enough energy to transmit to a neighbor node. Energy can also be conserved by improving the data-gathering and routing efficiency of the sensor nodes. The most challenging part of a sensor network with mobile sensors is routing of the mobile sensors. Because they have limited power, they cannot move great distances to cover the non-coverage areas, hence their movement must be limited locally. In this paper, we limit the relocation range of the mobile sensors up to 8 grid cells' radius in the grid area. By controlling the movements of the sensors in the network, we are conserving energy. There are several algorithms and procedures for routing mobile sensors and improving the sensor coverage, such as the local spiral search algorithm, random displacement algorithm, centralized algorithm, and distributed algorithm. Due to the overlap of sensors if multiple sensor nodes are collecting the same information, the network is expending energy needlessly. This paper demonstrates the random displacement of the mobile sensors during overlap technique which makes the sensor network more efficient by eliminating redundancy when relocating the overlapped sensors to non-coverage areas.

Over the past few years, much research has been performed in the sensor-coverage domain, and many new techniques have been proposed as solutions (Meguerdichian et al.). This project proposes a possible solution to the coverage problem. This paper demonstrates the random displacement of the mobile sensors during overlap technique which uses the random displacement algorithm. The random displacement of the mobile sensors during overlap tool was designed and implemented to provide a prototype of the sensor network. The tool has been tested with static and mobile sensor nodes randomly scattered in the sensor-network environment, with no overlapping sensors, resulting in considerable improvement in the coverage. The gain was calculated based on the sensor overlap.

The random displacement of the mobile sensors during overlap technique uses mobile sensors along with static sensors to take care of the coverage issue and to improve the performance for the sensor network. There are various reasons for including mobile sensors with static sensors. If there is an overlap between the mobile sensor and the static sensor, the mobile sensor can relocate to a non-coverage area. If there is an overlap between two mobile sensors, one of the mobile sensors can relocate to a non-coverage area. With the random displacement of the mobile sensors during overlap technique, there is no need to deploy additional sensors to cover a given area when there are mobile sensors. Furthermore, if a static or a mobile sensor dies in a high-risk area, the nearest low-risk area mobile sensor can relocate to the required area in order to continuously monitor the high risk area. This technique ensures continuous coverage and increased performance of the sensor network. The random displacement of the mobile sensors during overlap technique is further discussed in the next chapter.

4. RANDOM DISPLACEMENT OF MOBILE SENSORS DURING OVERLAP

The coverage issue and maximizing the coverage can be achieved by using random displacement of the sensors when there is any overlap between sensors. The technique will improve the sensor coverage and also improve the quality of service.

This procedure applies to a sensor network with both static and mobile sensors. Because of cost restrictions, a sensor network cannot have all mobile sensors. A cost-efficient sensor network is one with both static and mobile sensors where the static sensors outnumber the mobile sensors. The random displacement technique for overlap was chosen over other techniques because of its simplicity. Moreover, there is no need to save the sensor information.

The sensors are deployed in a given area with a combination of static and mobile sensors. Once they are deployed, if there is an overlap between a static and a mobile sensor or between two mobile sensors then the overlapped mobile sensor will move to a new location. By moving the mobile sensors which overlap with other mobile sensors or static sensors, a better coverage gain can be achieved and result in improved quality of service. The random displacement of the mobile sensor with overlap technique starts by moving the mobile sensor with maximum overlap. The mobile sensor will move to an empty grid cell which needs coverage. The movement of the mobile sensor is restricted by the number of steps. If we do not limit the movement, then the mobile sensor's power will be drained at an increasing level. If a grid is partially or fully covered by a sensor, then the information will be recorded, and the mobile sensor looking for an empty spot will be notified. Based on the information, the mobile sensor will move to the empty grid cell, and once again, the entire field would be tested for any overlap. The sensor movement is limited within 8 grid cell areas in order to achieve maximum coverage with minimum battery life. If there is no empty grid cell within the 8-grid radius area, then the overlapping sensor will not move. This process would continue until all the overlapping mobile sensors are moved to an empty grid cell.

Consider this example (Figure 1): If the sensor overlapping occurs in grid cell 0 (highlighted yellow), then the mobile sensor can randomly move to any non-covered grid cell within the radius of 8 grid cells (highlighted green).

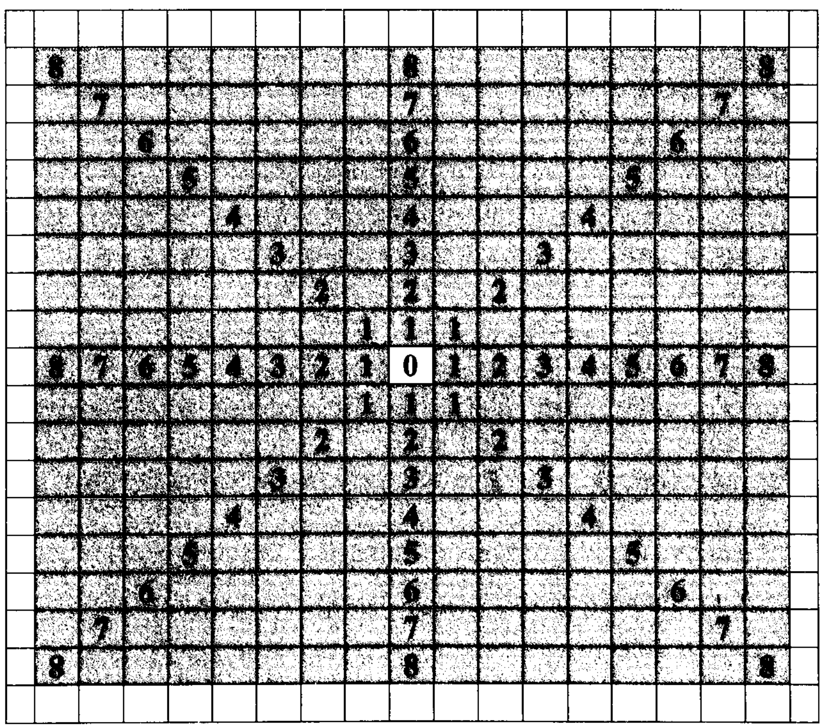

Figure 1. Grid Diagram.

The random displacement procedure is efficient because it works step by step, moving the mobile sensor with the most overlap to the grid cell with no coverage. Once this step is done, another reading is taken to determine the coverage of the grids. Then, the next-most overlapped sensor is moved to the grid cell with no coverage. No grid cell will be occupied by two mobile sensors. This technique is repeated until all the overlapped sensors move to uncovered grid-cell areas. This technique drastically enhances the quality of service, thereby maximizing the sensor coverage in a given area. Further descriptions of the application and the high-level design are discussed in the following chapters.

This procedure was tested with different test data, and each time, it produced an increased coverage gain, hence proving that this procedure is instrumental in improving the sensor coverage for a particular area. These results for different test cases are provided later in the paper.

5. HIGH-LEVEL DESIGN

The Unified Modeling Language (UML) is becoming widely used for both databases and software modeling. UML includes diagrams for use cases, static structures (class and object diagrams), behavior (state-chart, activity, sequence, and collaboration diagrams), and implementation (component and deployment diagrams).

5.1. Class Diagram

The class diagram is a static-structure diagram that describes the entire structure of a system by showing the system's classes, attributes of the classes, and the relationships between the classes. Classes are depicted as boxes with three sections: the top section indicates the name of the class; the middle section lists the attributes of the class; and the bottom section lists the methods.

The prototype (Figure 2) was designed with a total of 11 classes and 6 base classes. Each class is isolated from each other, and there is no dependency between any of the classes. Every class has its own functionality and performs different actions with no reusability of other classes.

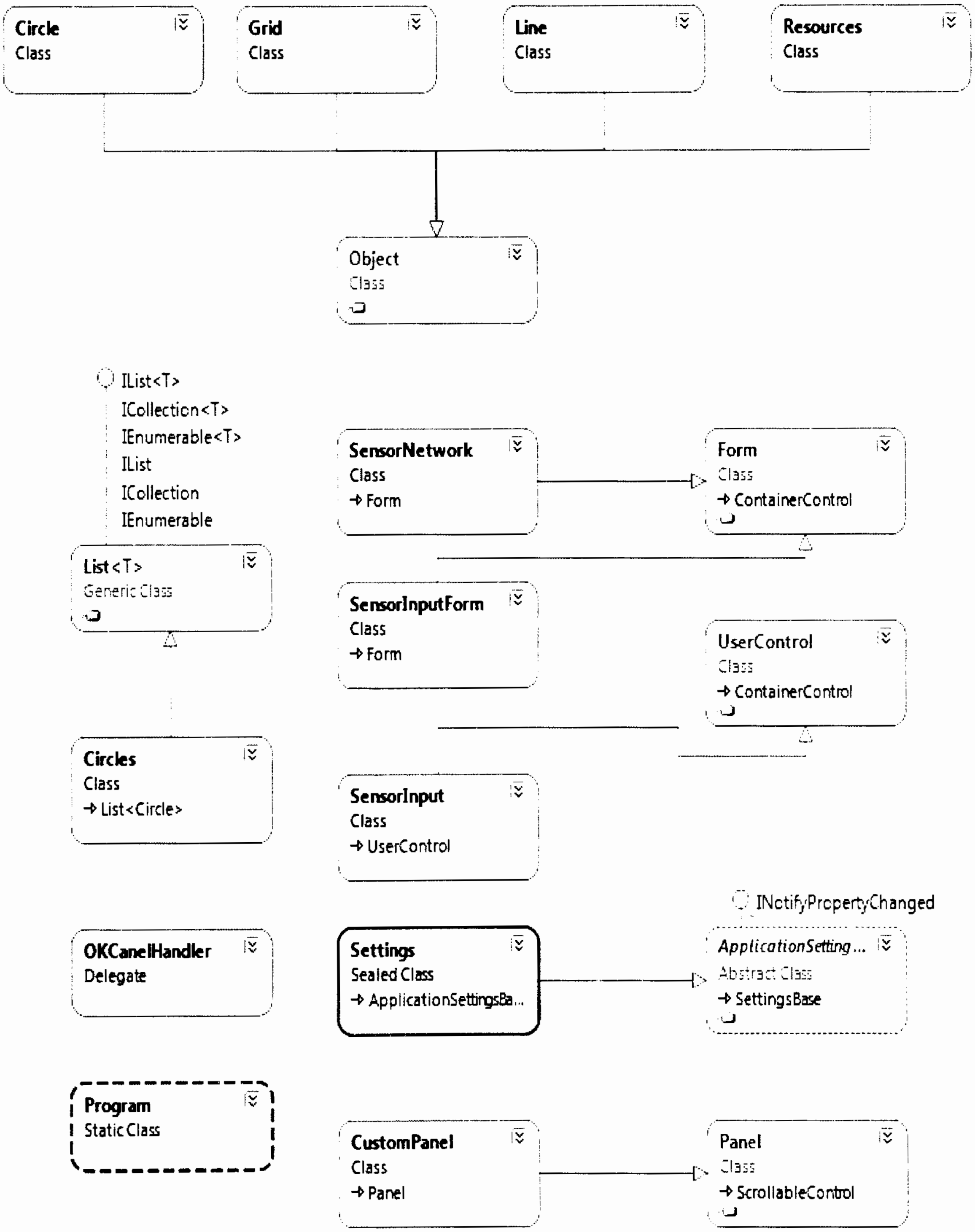

Figure 2. Class Diagram.

5.2. Sequence Diagram

The sequence diagram is one type of Unified Modeling Language (UML) diagram. It is an interactive diagram that shows the order of the processes that are operating with each other. There are total of three sequence diagrams that were developed for the prototype of the random displacement technique: Deploy sequence diagram, Move sequence diagram, and Move all sequence diagram.

5.2.1. Deploy Sequence Diagram

Once the input data are provided, the deploy operation can be performed anytime. Once we deploy the sensors, previously deployed sensors will be removed, and new sensors will be deployed based on the input provided by the user.

When sensors are deployed in the field, the entire area will be cleared, and static sensors will be deployed, followed by the mobile sensors. The displacement will be uncontrolled; hence the sensors can be located anywhere in the given field. Upon complete deployment, the sensor-overlap area and sensor field information, such as the total area, coverage area, and non-coverage area, are accumulated.

Area calculation will be performed based on the information collected from the mobile and static sensor locations as well as their overlap areas. Overlap for the mobile sensor area will be calculated first, followed by the static overlap area. Based on the information collected, the coverage area of the sensors and non-coverage area of the sensors will be calculated. The overlapped mobile sensors are set to move to new locations with the collected information about the coverage and non-coverage areas. Figure 3 explains the sequence diagram for the deploy sensor mechanism.

5.2.2. Move Sequence Diagram

The Move action is performed to re-locate the overlapped mobile sensors to the non-covered region. At first, the overlap-area information is collected, and a decision is made based on the percentage of the overlap area between each mobile-mobile and mobile-static sensor. The mobile sensor with maximum overlap percentage is selected first for random displacement. Within the 8-grid radius, the overlapped mobile sensor can move randomly to any non-covered region. With the overlapped mobile sensor at a new location, the entire area of the sensors is re-calculated. The re-calculation is performed using the update grid process which gathers all the information about the coverage and non-coverage of the sensors. Finally, the gain percentage is calculated based on the collected information from the updated coverage and non-coverage areas. The sequence diagram of the Move operation is depicted in Figure 4.

Calculation of the gain percentage is represented in the formula below where the UpdateCovergaeArea is the coverage area of the sensors calculated after randomly moving the overlapped mobile sensor to new location and the InitialCoverageArea is the coverage area of the sensors calculated before moving the overlapped mobile sensors.

*Gain = [(UpdatedCoverageArea – InitialCoverageArea) / UpdatedCoverageArea) * 100]*

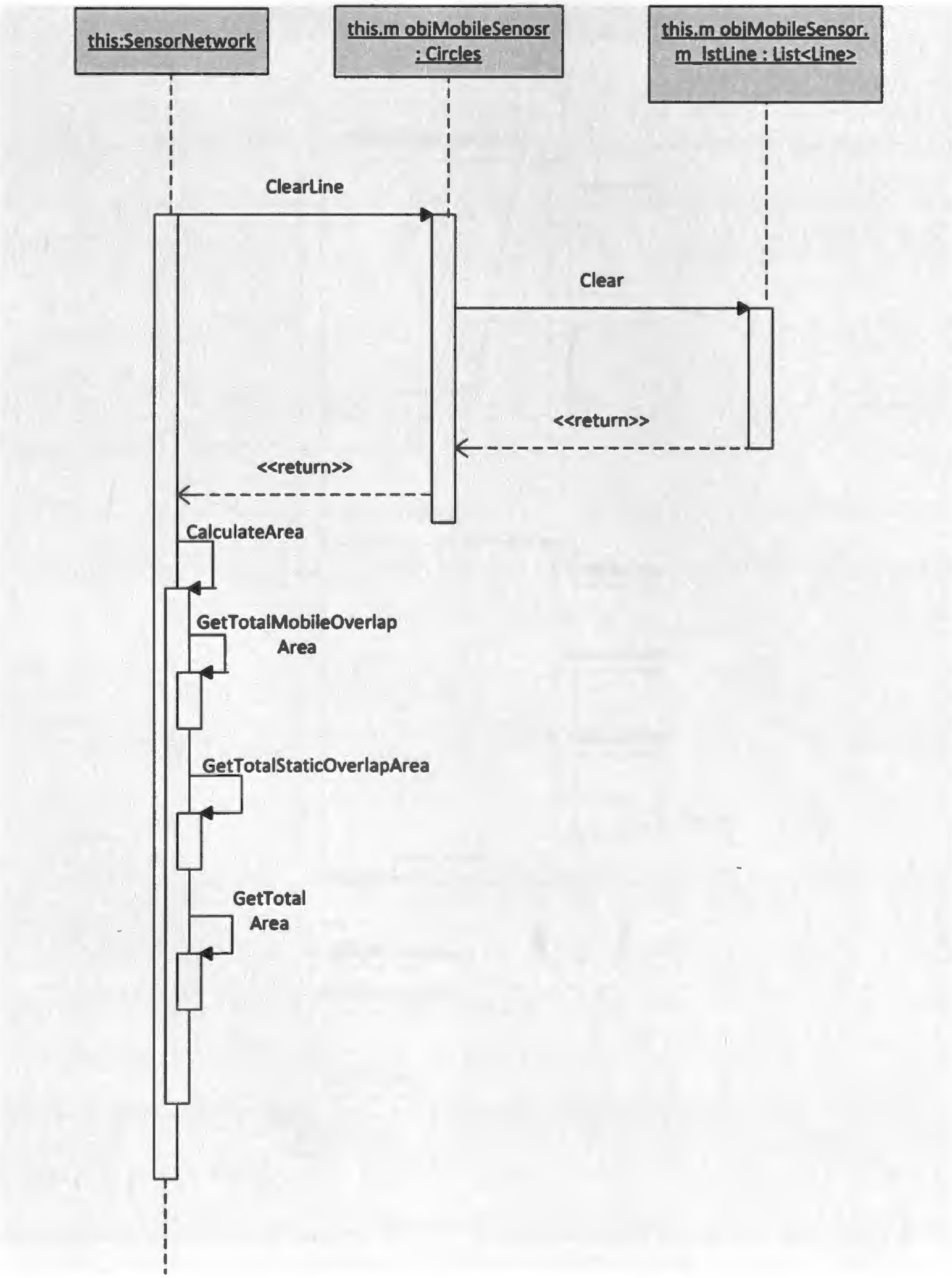

Figure 3. Deploy Sequence Diagram.

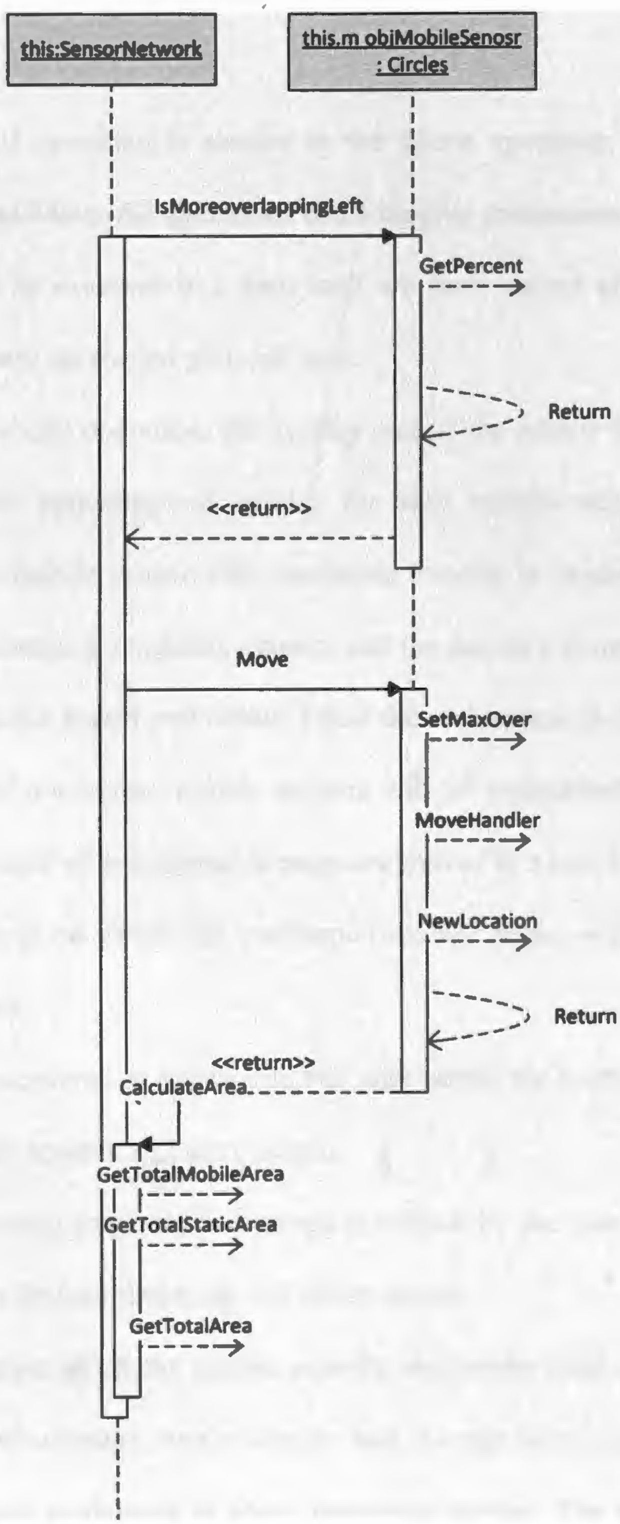

Figure 4. Move Sequence Diagram.

5.2.3. Move-All Sequence Diagram

The Move-All operation is similar to the Move operation. The only difference between the Move and Move-All operations is the looping mechanism. With Move-All, the Move operation will be executed in a loop until we have moved all possible overlapped mobile sensors to a new uncovered grid-cell area.

With the Move-All operation, the overlap area of the sensor field is calculated, and information about the percentage of overlap for each mobile-mobile and mobile-static sensor is saved. The mobile sensor with maximum overlap is randomly moved to a new location. The new location is randomly chosen, and the decision is made based on the non-coverage area within the 8-grid cell radius. Once the re-location is completed, the second highest percentage of overlapped mobile sensors will be considered for re-location. This process is continued until all overlapped sensors are moved to a new location.

The two criteria on which the overlapped mobile sensor will not move to a new location are as follows:

- If there is no non-covered or empty grid cell area within the 8-grid cell radius, then the overlapped mobile sensors will not relocate.
- If a minimum overlap threshold percentage is defined by the user, then any overlapped sensors below the defined threshold will not re-locate.

Upon re-location of all the mobile sensors, the sensor field is updated with latest information about the coverage, non-coverage, and overlap areas. Calculation of the grid percentage is the same as defined in Move operation section. The sequence diagram for Move-All sensors is depicted in Figure 5.

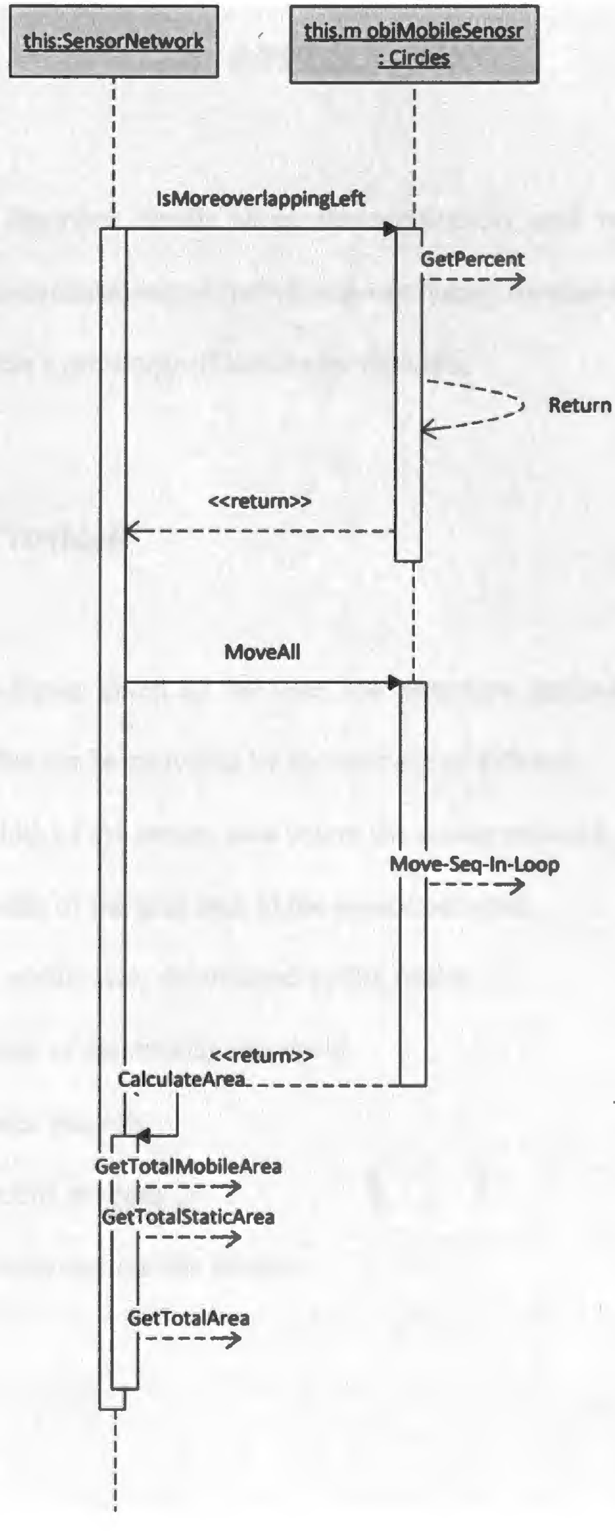

Figure 5. Move - All Sequence Diagram.

6. APPLICATION

This chapter describes details about the application used to generate the sensor network. The random displacement of mobile sensors during overlap tool was designed and implemented to provide a prototype of the sensor network.

6.1. Inputs Provided

Based on the inputs given by the user, the prototype generates a sensor network. The possible inputs that can be provided by the user are as follows:

- The height and width of the sensor area where the sensor network is to be created.
- The height and width of the grid area in the sensor network.
- The homogenous sensor size, determined by the radius.
- The percentage limit of the overlap threshold.
- The number of static sensors.
- The number of mobile sensors.
- The color of the static and mobile sensors.

6.2. Color Coding

The color of the grid-cell background, static sensors, and mobile sensors are chosen by the user. The colors can be provided either while entering the data or in the combo box. The colors can be changed for each test case.

6.3. The Task Bar

6.3.1. 'Data' Tab

The 'Data' tab allows the user to enter the inputs required for the application. Default values are provided for the user.

6.3.2. 'Deploy Sensor' Tab

Upon entry of the required sensor data, the sensor network prototype will be generated by the 'Deploy Sensor' tab. The user's inputted value for the number of static sensors and the number of mobile sensors will be randomly positioned in the grid area.

6.3.3. 'Move' Tab

By using the 'Move' tab, the overlapped mobile sensors will randomly be positioned to a non-coverage area. Each move is determined by the maximum overlaps. The maximum overlapped sensors are moved first to non-covered areas, followed by the next-highest overlapped sensors. If the user defines an overlapping threshold limit, then the overlapped mobile sensors which are below the threshold value will not move to a non-coverage area.

6.3.4. 'Move All' Tab

By using the 'Move All' tab, the overlapping mobile sensors will be positioned randomly at the same time. If the user defines an overlapping threshold limit, then the overlapped mobile sensors which are below the threshold value will not move to a non-coverage area.

6.4. The Combo Box

6.4.1. 'Total Area' Field

The 'Total Area' field displays, in pixels, the total area of the sensor network. The total area data is obtained from the sensor area input. The sensor area height and width multiplied together gives the total area of the sensor field.

6.4.2. 'Coverage Area' Field

The 'Coverage Area' field displays the total coverage area of the sensor network, for both the static and mobile sensors, in pixels. The calculated coverage area is then used to obtain the non-coverage area of the sensor field.

6.4.3. 'Non-Coverage Area' Field

The 'Non-Coverage Area' field displays, in pixels, the total area which is not covered by either static or mobile sensors. The non-coverage area is obtained by subtracting the total area by coverage area of the sensor field.

6.4.4. 'Overlap Area' Field

The 'Overlap Area' field displays, in pixels, the total overlap area between the sensors. The overlap area is calculated between static-static and static-mobile sensors. The percentage of the overlapped area can be obtained in the overview field. Refer 'Overview' field section for more details on overlapped percentage data.

6.4.5. 'Gain' Field

The corresponding coverage gain for each move can be measured and is displayed in the 'Gain' field. The total coverage gain of the sensor area, as a percentage, can be calculated by {[(Coverage Area – Initial Coverage Area) / Coverage Area] * 100}

6.4.6. 'Overview' Field

The 'Overview' field provides the user a general description of the sensor area. For every move, it provides details about the gain of the coverage sensors. If the mouse is moved over the overlapped sensor area, the overlapping percentage is displayed in the 'Overview' field.

7. VISUALIZATION

This chapter of the paper will cover the visualization of the sensor network and describe how the static and mobile sensors are positioned across the grids without any overlap. Figure 6 is data-entry screen. The user can enter the required data to run the prototype. Data for the Sensor Area; Grid Area; sensor size; threshold overlapping percentage; and the number of static and mobile sensors, along with their respective color denotions, can be entered in the form. Upon completion of the form, the 'OK' button is used to generate the sensor network. The 'Cancel' Button is used to exit the setting form without making any changes to the default values.

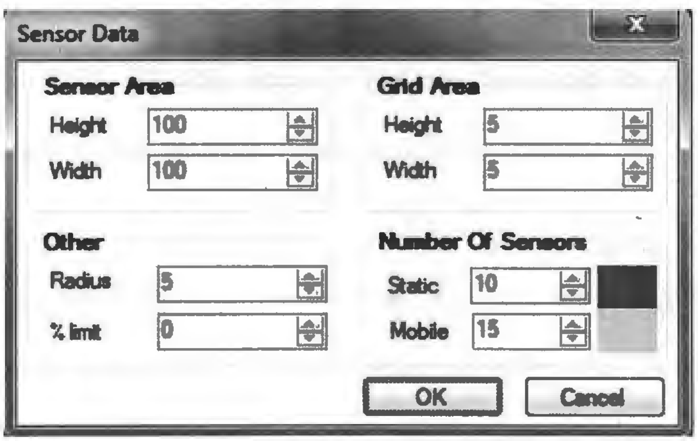

Figure 6. Setting Form for Data Entry.

Figure 7 shows the overlaps among the sensors in the network. This screen is generated upon deploying the sensors. The 'Move' tab can be used to reposition the mobile sensors to the non-coverage areas so that there is no overlap. The most-overlapped sensors are moved first, followed by the next-most overlapped sensors, using the random displacement technique algorithm. This process is continued until all the overlapped sensors above the overlapping threshold percentage have been repositioned to the non-coverage grid-cell area. Upon positioning the mouse over the overlapped sensors, that particular overlap area will be displayed in the Overview field. The 'Move All' tab can also be used to reposition the overlapped sensor to the non-coverage grid-cell area. The only difference between the 'Move' and 'Move All' tabs is that the 'Move' tab repositions the overlapped sensors one-by-one, whereas the 'Move All' tab repositions all overlapped sensors at once. The combo box also displays the Total Area field, Coverage Area field, Non-Coverage Area field, Overlap Area field, Gain field, and the Overview field. The respective color denotions for the Grid, Static, and Mobile sensors are also displayed in the combo box.

Figure 8 is a sensor network that does not have overlaps between the sensors. All the overlapped mobile sensors have been repositioned to the non-coverage grid area. The dotted line represents the relocation of the overlapping sensors, from the initial location to the new location. The combo box also displays the Total Area field, Coverage Area field, Non-Coverage Area field, and the Overlap Area field. Once all the overlapped sensors are relocated, the final gain percentage is displayed in the Gain field. The random displacement technique procedure is used for different test cases, and the corresponding gain percentage can be calculated. Once all the mobile sensors are repositioned across the grid area so that none are overlapping, a 'Sensor Result' pop-up appears with the following message: "There was no overlap in the sensor coverage." This message is an indication that the random displacement technique algorithm was sucessfully implemented to generate a sensor network with no overlapping sensors.

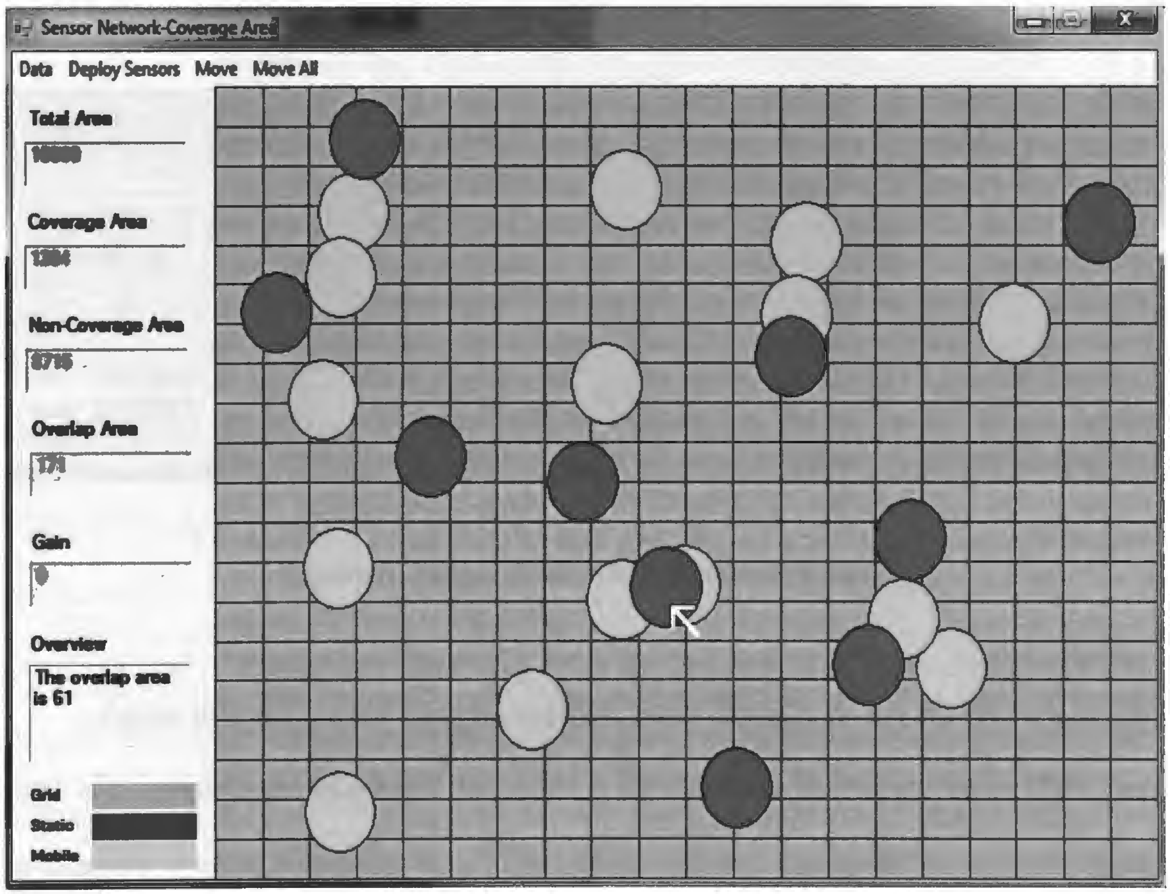

Figure 7. Sensor Network with Overlaps.

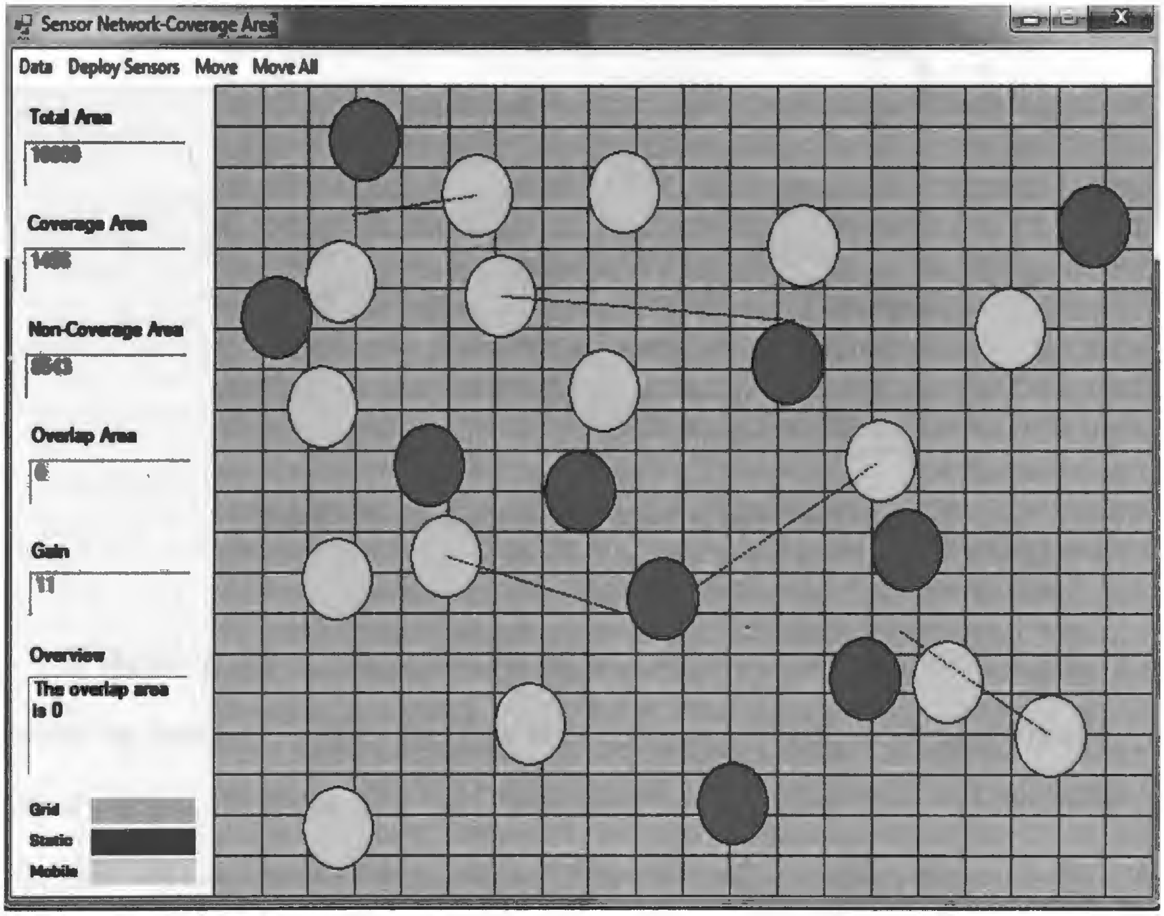

Figure 8. Sensor Network Without Overlaps.

Figure 9 is the setting form for the data entry required to run the prototype. This form is a sample data form with the overlapping threshold percentage set as 25%. Any overlapping sensors with more than 25% overlap will be repositioned to a non-coverage area. If the overlapping sensors are less than the 25% overlap threshold, they will not relocate to a non-coverage area. Upon completion of the form, the "OK" button is used to generate the sensor network. The "Cancel" button is used to exit the setting form without making any changes to the default values.

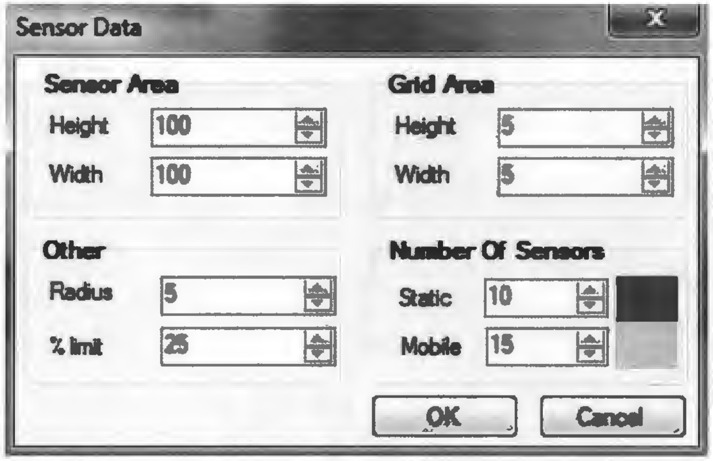

Figure 9. Data Entry with 25% Overlap Threshold.

Figure 10 displays the sensor grid with the deployed static and mobile sensors. The mouse has been positioned on top of the overlapping sensors with more than 25% overlap. When clicking the Move Tab, the overlapping sensors greater than the 25% overlap threshold will move to non-coverage areas. If the overlapping sensors are less than the 25% overlap threshold, they will not relocate. The Overlap Area field displays a value of 221 for this test case.

Figure 11 displays the sensor grid when the Move All tab is clicked. The Overlap Area is now 127 for this test case. The gain obtained is 7. The dotted lines indicate the path of the moving overlapped sensors from the initial location to the new location. As noted in the Figure 11, the overlapped sensors below the overlap threshold of 25% have not been repositioned. Hence, the Overview field displays “The overlap area is 0.”

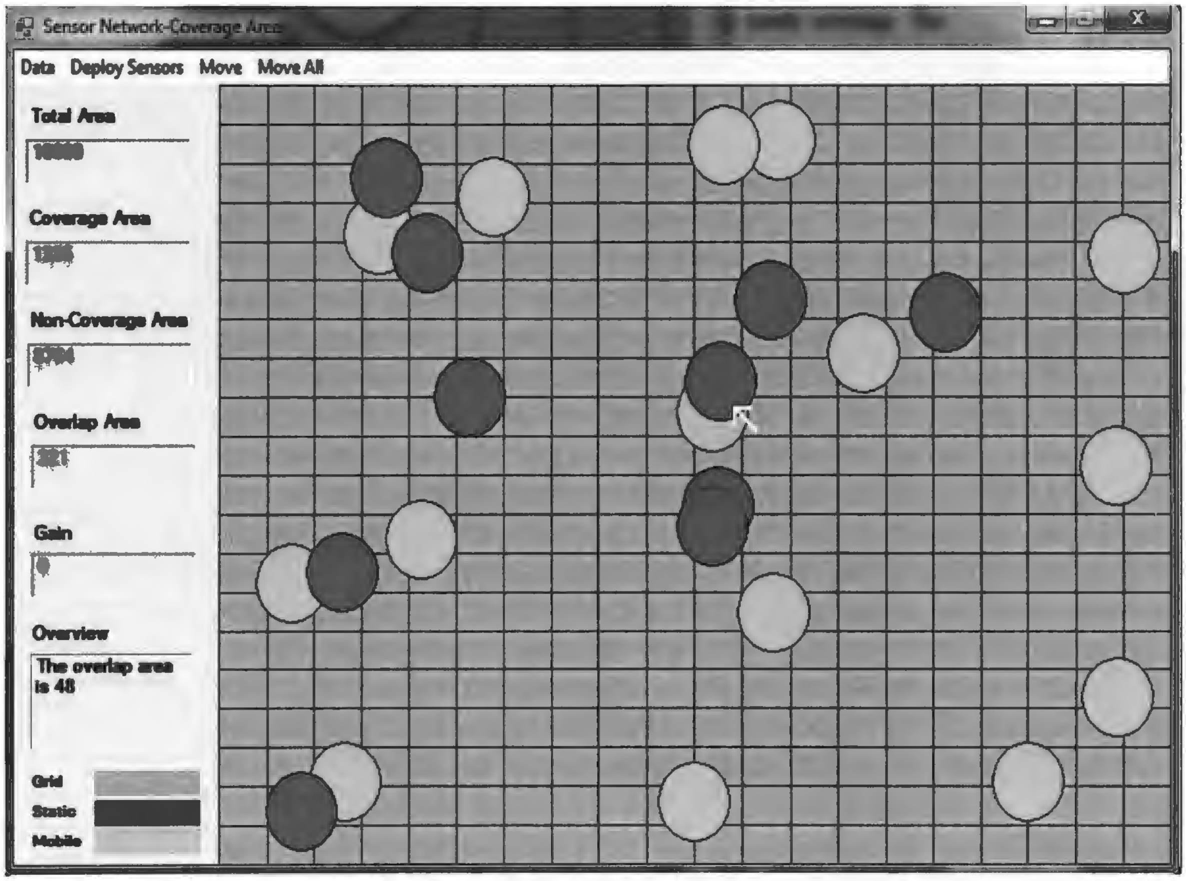

Figure 10. Sensor Network with Overlap > 25%.

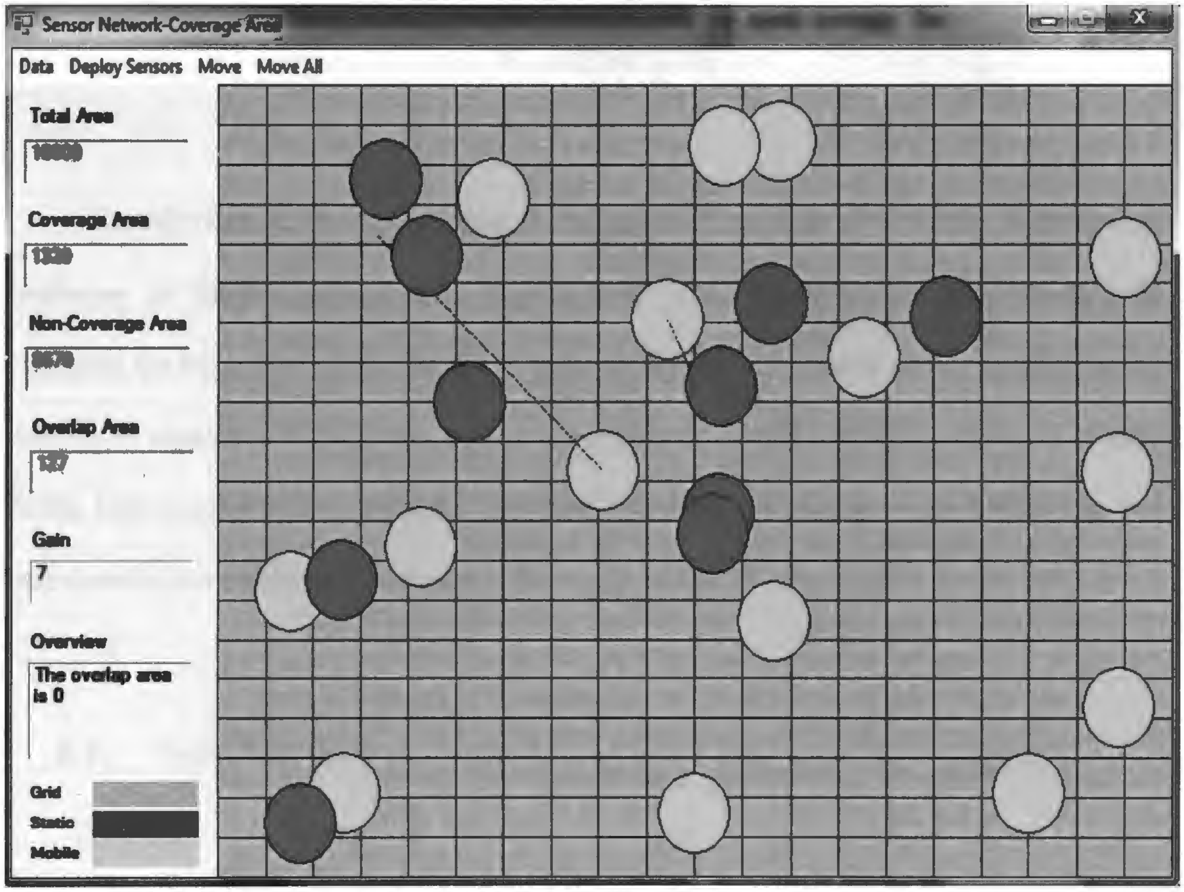

Figure 11. Relocation of Overlapping Sensors > 25%.

8. RESULTS

In this chapter, the performance of the tool developed for the random displacement technique of mobile sensors will be validated. Different scenarios were created by changing the input data of the sensors. Sensor area, grid area, radius of the sensor, and the number of sensors to be deployed were the key elements while generating different sensors fields. Below are different test cases that were validated; the average mean of the gain was calculated at the end of each test case.

8.1. Test Case 1

As seen in Table 1, for the first test case, a sensor area of 100x100 (in pixels), grid size of 5x5 (in pixels), sensor radius size of 5, 15 static sensors, and 15 mobile sensors were given as input. With the given input data, the sensors were deployed 20 times, resulting in the sensors being displaced at random positions. Gain was calculated based on the overlap of the sensors. The highest gain was noted at 20%, and the lowest gain was reported as 4% during execution of the test case. Average gain resulted at 12.5%.

8.2. Test Case 2

As seen in Table 2, for the second test case, a sensor area of 200x200 (in pixels), grid size of 10x10 (in pixels), sensor radius size of 10, 25 static sensors, and 25 mobile sensors were given as input. The highest gain resulted at 44%, and the lowest gain resulted at 12%. The average gain of the sensors was reported as 24%.

8.3. Test Case 3

As seen in Table 3, for the third test case, a sensor area of 300x300 (in pixels), grid size of 15x15 (in pixels), sensor radius size of 15, 25 static sensors, and 25 mobile sensors were given as input. The highest gain resulted at 32%, and the lowest gain resulted at 13%. The average gain of the sensors was reported as 20.30%.

Average gain generated by different input data for the random displacement technique was always higher than the other algorithms, such as the Local Spiral Search algorithm (Katib). With the Local Spiral Search algorithm, the highest average gain was noted as 6.5% and lowest average gain was noted as 4.1%, whereas with the random displacement technique, the highest average gain was noted as 44%, and the lowest average gain was noted as 4%. Figure 12 explains the comparison of gain data among Test Cases 1, 2, and 3.

Table 1. Test Case 1

Total Area	10000		
Sensor Area	100x100(in pixel)		
Grid Size	5x5(in pixel)		
Sensor Radius Size	5		
Number of Sensors			
	Static	15	
	Dynamic	15	

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
1650	8349	97	1737	8262	10	5
1361	8638	386	1705	8294	42	20
1421	8578	326	1608	8391	139	11
1519	8480	228	1679	8320	68	9
1325	8674	422	1670	8329	77	20
1357	8642	390	1690	8309	57	19
1578	8421	169	1714	8285	33	7
1439	8560	308	1670	8329	77	13
1414	8585	333	1651	8348	96	14
1319	8680	428	1667	8332	80	20
1576	8423	171	1732	8267	15	8
1473	8526	274	1706	8293	41	13
1557	8442	190	1691	8308	55	7
1484	8515	262	1597	8404	150	7
1360	8639	387	1657	8342	90	17
1489	8510	258	1721	8278	26	13
1542	8457	205	1747	8252	0	11
1668	8331	79	1747	8252	0	4
1371	8628	376	1642	8357	105	16
1260	8739	487	1514	8485	233	16
					Average	12.5

Table 2. Test Case 2

Total Area	40000	
Sensor Area	200x200(in pixel)	
Grid Size	10x10(in pixel)	
Sensor Radius Size	10	
Number of Sensors		
	Static	25
	Dynamic	25

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
8796	31203	2855	10817	29182	834	18
7035	32964	4616	10488	29511	1163	32
7168	32831	4483	10766	29233	885	33
8573	31426	3079	10962	29037	689	21
7782	32217	3870	10227	29772	1425	23
8185	31814	3466	10571	29428	1081	22
5891	34108	5760	10568	29431	1084	44
8241	31758	3410	10707	29292	944	23
7140	32859	4511	10269	29730	1382	30
8404	31595	3247	11294	28705	357	25
7364	32635	4287	10467	29532	1184	29
7666	32333	3985	10869	29130	782	29
8304	31695	3347	10718	29281	933	22
9003	30996	2649	11364	28635	287	20
8105	31894	3546	10474	29525	1177	22
8146	31853	3505	10541	29458	1110	22
8539	31460	3113	10495	29504	1156	18
9791	30208	1860	11505	28494	146	14
8771	31228	2880	11174	28825	477	21
9383	30616	2268	10765	29234	886	12
					Average	24

Table 3. Test Case 3

Total Area	90000	
Sensor Area	300x300(in pixel)	
Grid Size	15x15(in pixel)	
Sensor Radius Size	15	
Number of Sensors		
	Static	25
	Dynamic	25

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
17817	72182	8399	23710	66289	2506	24
20153	69846	6063	25408	64591	808	20
19188	70811	7029	23243	66756	2973	17
21337	68662	4880	25092	64907	1124	14
19518	70481	6698	24729	65270	1487	21
18042	71957	8175	24311	65688	1905	25
20881	69118	5335	24461	65538	1756	14
19932	70067	6284	25117	64882	1099	20
21165	68834	5051	25053	64946	1163	15
17177	72822	9039	25307	64692	909	32
20342	69657	5874	25368	64631	848	19
19644	70355	6572	24597	65402	1619	20
22973	67026	3244	26152	63847	65	12
18686	71313	7530	27726	67223	3441	17
21260	68739	4957	24461	65538	1755	13
18548	71451	7669	24754	65245	1462	25
16874	73125	9342	24680	65319	1537	31
20520	69479	5696	25614	64385	602	19
18753	71246	7463	23320	66679	2896	19
16724	73275	9492	23558	66441	2658	29
					Average	20.3

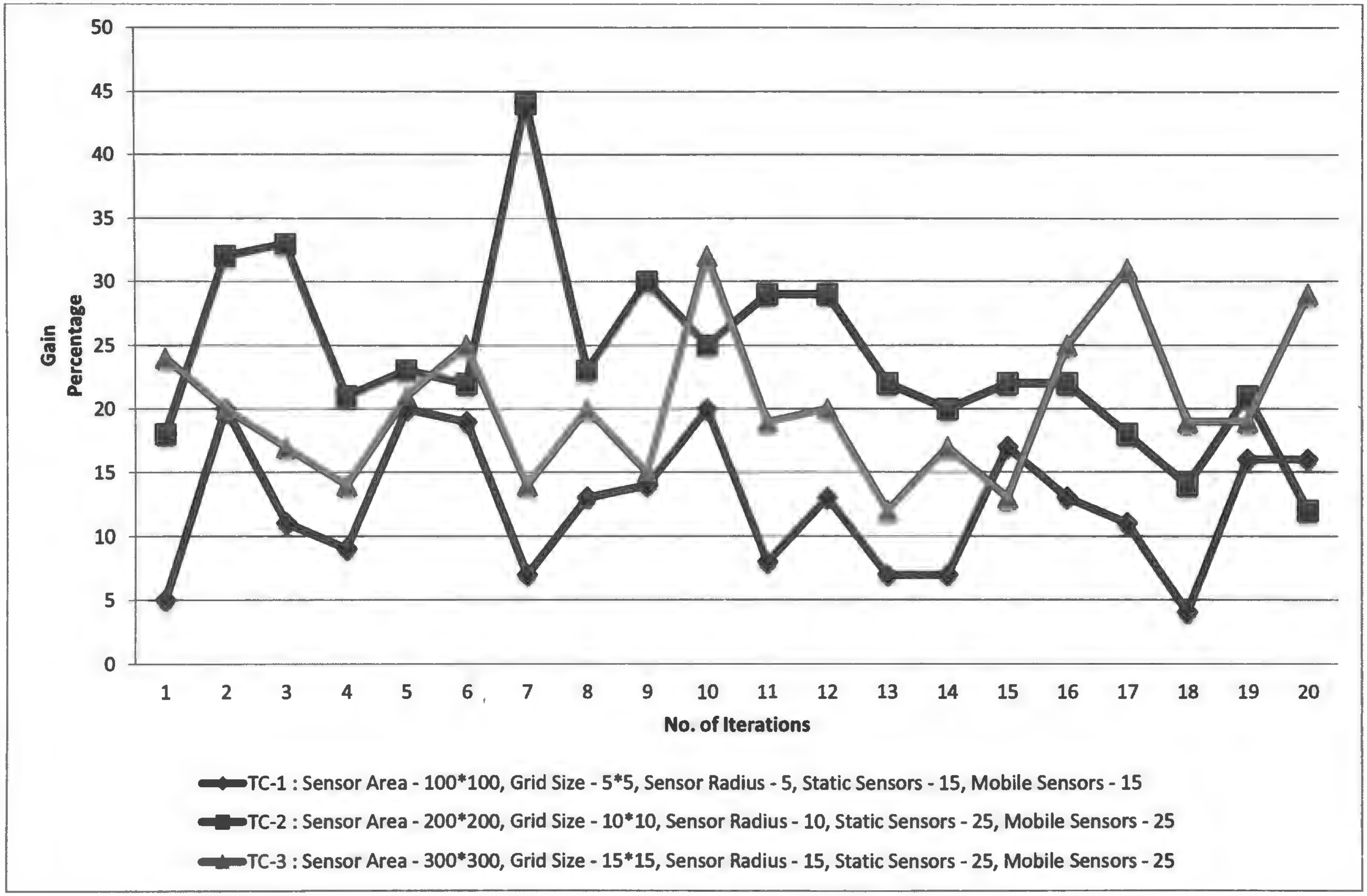

Figure 12. Comparison of Gain Data from Test Cases 1, 2 and 3.

8.4. Test Case 4

The fourth test case was executed to cover the gain improvisation. The total number of static sensors is a fixed number, and the total number of mobile sensors was increased for every set of iterations.

All fields except the number of mobile sensors were kept as fixed data. The sensor area of 100x100 (in pixels), grid size of 5x5 (in pixels), sensor radius size of 3, and 100 static sensors were given as fixed input. The total number of mobile sensors was increased in an incremental order of 5, 10, 15, etc.

As seen in Figure 13 and Table 4, when 5 was the total number of mobile sensors, the average gain was 2. When 10 was the total number of mobile sensors, the average gain resulted in 4. The average gain resulted in an incremental curve on an order of 2.

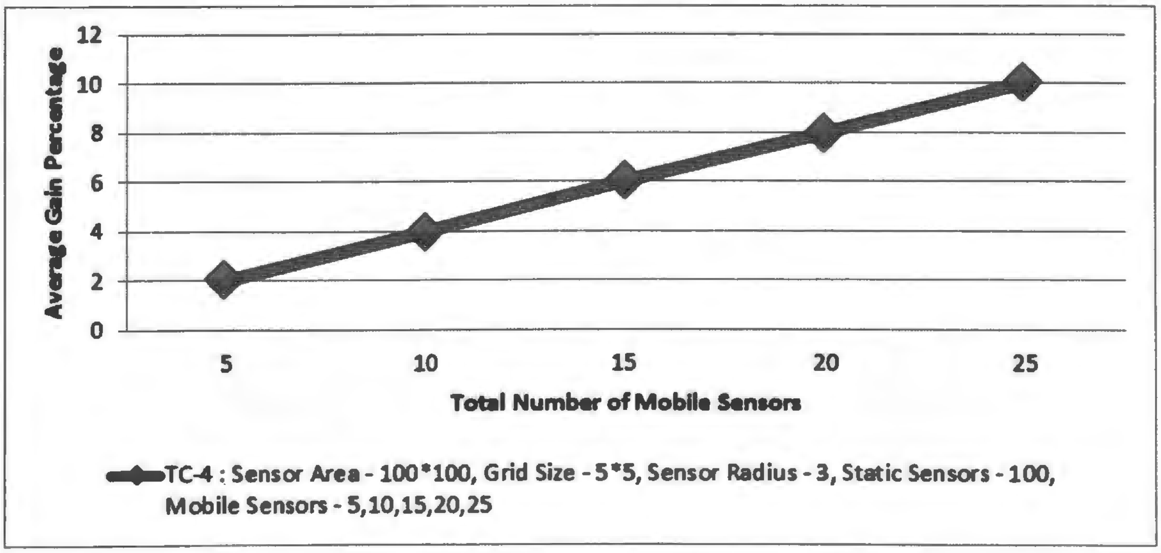

Figure 13. Gain Coverage with Incremental Mobile Sensors.

Table 4. Test Case 4

Total Area	10000
Sensor Area	100x100(in pixel)
Grid Size	5x5(in pixel)
Sensor Radius Size	3
Number of Sensors	
Static	100
Dynamic	5

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
1779	8220	423	1811	8188	390	1
1673	8326	528	1761	8238	440	4
1667	8332	534	1730	8269	471	3
1834	8165	367	1854	8145	347	1
1729	8270	472	1749	8250	452	1
1753	8246	448	1782	8217	420	1
1617	8382	584	1662	8337	539	2
1788	8211	414	1858	8141	343	3
1736	8263	465	1764	8235	437	1
1823	8176	378	1842	8157	359	1
					Average	2

Total Area	10000
Sensor Area	100x100(in pixel)
Grid Size	5x5(in pixel)
Sensor Radius Size	3
Number of Sensors	
Static	100
Dynamic	10

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
1802	8197	504	1907	8092	400	5
1879	8120	427	1952	8047	354	3
1733	8266	573	1854	8145	452	6
1876	8123	430	2000	7999	306	6
1843	8156	463	1881	8118	426	2
1784	8215	522	1876	8123	430	4
1847	8152	459	1928	8071	379	4
1882	8117	424	1949	8050	357	3
1861	8138	445	1912	8087	394	2
1976	8023	330	2032	7967	275	2
					Average	4

Total Area	10000
Sensor Area	100x100(in pixel)
Grid Size	5x5(in pixel)
Sensor Radius Size	3
Number of Sensors	
Static	10
Dynamic	15

Coverage Area (Before Applying Algorithm) [in pixel]	Non Coverage Area (Before Applying Algorithm) [in pixel]	Overlap Area (Before Applying Algorithm) [in pixel]	Coverage Area (After Applying Algorithm) [in pixel]	Non Coverage Area (After Applying Algorithm) [in pixel]	Overlap Area (After Applying Algorithm) [in pixel]	Gain (%)
1957	8042	454	2038	7961	373	3
1814	8185	597	1978	8021	433	8
1862	8137	549	2003	7996	408	7
1859	8140	552	1939	8060	472	4
1941	8058	470	2053	7946	358	5
1891	8108	520	1960	8039	451	3
1790	8209	621	1895	8104	516	5
1945	8054	466	2042	7957	369	4
1810	8189	601	1986	8013	425	8
1754	8245	657	1869	8130	542	6
					Average	6

9. CONSTRAINTS

The following are constraints of the Random Displacement Technique:

- Considering the battery health of the mobile sensor, the sensor can move up to 8 steps in any direction at any given point of time.
- No further movement can be achieved once minimum overlap is attained.

10. CONCLUSION

In addition to tracking and deployment, one fundamental issue in sensor networks is the coverage problem, which reflects how well a sensor network is monitored or tracked by sensors. The better the coverage, the better the sensor network performance is in a given region. This project proposed a possible solution for the coverage problem. This paper demonstrated the random displacement of the mobile sensors during overlap technique which was designed and implemented in order to provide a prototype of the sensor network. Achieving maximum coverage using random displacement in a limited area can be accomplished only when we have more mobile sensors in addition to the static sensors deployed in the sensor field.

The random displacement technique was used to reposition any overlapping mobile sensors with static or mobile sensors in order to enhance the quality of service and to increase the coverage in a given area. The gain was calculated based on the overlap of the sensors. The user was given an option to define the overlap threshold limit. Sensors below the overlap threshold value did not move to a new location. In order to achieve maximum coverage with minimum battery life, the movement of the sensors was restricted within an 8-grid cell radius. Sensors did not move to a new location if there was no empty cell within the 8-grid radius area. The overlapped mobile sensor moved randomly to a non-coverage area within the 8-grid radius area. Results of various test cases and their gains were discussed in the paper. There was a considerable increase in the sensor coverage area when the random displacement of the mobile sensors during overlap technique was used.

REFERENCES

Batalin, Maxim A., and Gaurav S. Sukhatme. *Coverage, Exploration and Deployment by a Mobile Robot and Communication Network*. Tech. Proceedings of the International Workshop on Information Processing in Sensor Networks. Palo Alto Research Center (PARC), Apr. 2003. Web. May 2011. <http://cres.usc.edu/pubdb_html/files_upload/27.pdf>.

Camtepe, Seyit A., and B¨ulent Yener. *Key Distribution Mechanisms for Wireless Sensor Networks: A Survey*. Tech. TR-05-07, Department of Computer Science, Rensselaer Polytechnic Institute, 23 Mar. 2005. Web. Aug. 2011. <http://www.cs.rpi.edu/research/pdf/05-07.pdf>.

Cardei, Mihaela, and Jie Wu. *Energy-Efficient Coverage Problems in Wireless Ad-Hoc Sensor Networks*. Elsevier Properties, Dec. 2007. Web. May 2011. <www.Sciencedirect.com>.

Dhillon, Santpal S., Krishnendu Chakrabarty, and S. S. Iyengar. *Sensor Placement for Grid Coverage Under Imprecise Detections*. Tech. Proceedings of the Fifth International Conference on Information Fusion, 2002. Web. Jan. 2011. <http://www.csc.lsu.edu/~iyengar/final-papers/isif.pdf>.

Du, Xiaojiang (James), Ming Zhang, Kendall E. Nygard, Sghaier Guizani, and Hsiao Hwa Chen. "Self-Healing Sensor Networks with Distributed Decision Making." *International Journal of Sensor Networks* 2.5/6 (2007): 289-98.

Inderscience Publishers, 2007. Web. July 2011. <http://dl.acm.org/citation.cfm?id=1359146>.

Fan, Gaojun, and Shiyao Jin. "Coverage Problem in Wireless Sensor Network: A Survey."*Journal of Networks* 5.9 (2010). Academy Publisher, Sept. 2010. Web. Aug. 2011. <http://ojs.academypublisher.com/index.php/jnw/article/viewFile/050910331040/2124>.

Howard, Andrew, Maja J. Matarić, and Gaurav S. Sukhatme. "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks." *Autonomous Robots Special Issue on Intelligent Embedded Systems* 13.2 (2002): 113-26. 2002. Web. Mar. 2011. <http://www-robotics.usc.edu/~ahoward/pubs/howard_ar01.pdf>.

Huang, Chi-Fu, and Yu-Chee Tseng. *The Coverage Problem in a Wireless Sensor Network.* Tech. Proceedings of the Second ACM International Conference on Wireless Sensor Networks and Applications, Sept. 2003. Web. June 2011. <http://dl.acm.org/citation.cfm?id=941367>.

Katib, Faraz. "Local Spiral Search in Sensor Coverage Problems." Thesis. North Dakota State University, 2008. Print.

Kung, Yao. *Sensor Networking: Concepts, Applications, and Challenges*. Tech. Acta Automatica Sinic, Nov. 2006. Web. June 2011. <http://www.aas.net.cn/qikan/manage/wenzhang/060601.pdf>.

Li, Jason H., and Miao Yu. "Sensor Coverage in Wireless Ad Hoc Sensor Networks." *International Journal of Sensor Networks* 2.3/4 (2007): 218-29. 2007. Web. Feb. 2011. <http://dl.acm.org/citation.cfm?id=1359012>.

Liu, Benyuan, and Don Towsley. *A Study of the Coverage of Large-Scale Sensor Networks*. Tech. IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Oct. 2004. Web. Aug. 2011. <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1392188>.

Meguerdichian, Seapahn, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. *Coverage Problems in Wireless Ad-Hoc Sensor Networks*. Tech. INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. 2001. Web. Aug. 2011. <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=916633>.

Mekid, Samir. "Further Structural Intelligence for Sensors Cluster Technology in Manufacturing." Ed. Wilmar Hernandez. *Sensors* 6.6 (2006): 557-77. 2006. Web. June 2011. <http://www.mdpi.com/1424-8220/6/6/557/>.

Mulligan, Raymond, and Habib M. Ammari. "Coverage in Wireless Sensor Networks: A Survey." *Network Protocols and Algorithms* 2.2 (2010). Macrothink Institute, 13 Apr. 2010. Web. Apr. 2011. <http://www.macrothink.org/journal/index.php/npa/article/view/276/332>.

Poduri, Sameera, and Gaurav S. Sukhatme. *Constrained Coverage for Mobile Sensor Networks*. Tech. IEEE International Conference on Robotics and Automation, Apr.-May 2004. Web. June 2011. <http://cres.usc.edu/pubdb_html/files_upload/409.pdf>.

Pottie, Gregory J. *Wireless Sensor Networks*. Killarney: Information Theory Workshop (1998): 139-40. 22-26 June 1998.

Savvides, Andreas, Chih-Chieh Han, and Mani B. Strivastava. *Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors*. Tech. Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking MobiCom 01, 2001. Web. July 2011. <http://dl.acm.org/citation.cfm?id=381693>.

Shang, Yi, Wheeler Rumi, Ying Zhang, and Markus Fromherz. "Localization from Connectivity in Sensor Networks." *IEEE Transactions on Parallel and Distributed Systems* 15.11 (2004): 961-74. IEEE Computer Society, Oct. 2004. Web. Aug. 2011. <ieeexplore.ieee.org/iel5/71/29526/01339247.pdf>.

Sohrabi, Katayoun, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie. "Protocols for Self-Organization of a Wireless Sensor Network." *IEEE Personal Communications* 7.5 (2000): 16-27. IEEE Communications Society, Oct. 2000. Web. Jan. 2011. <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=878532>.

Tagne, Elie Fute, Fabrice Lauri, Abder Koukam, and Emmanuel Tonye. "The Coverage Problem in Wireless Sensor Networks by Holonic Multi-Agent Approach." *International Journal of Computing and ICT Research* 3.1 (2009): 32-41. IJCIR, June 2009. Web. Aug. 2011. <http://ijcir.org/volume3-number1/issue%2032-41.pdf>.

Tseng, Yu-Chee, Shih-Lin Wu, Wen-Hwa Liao, and Chih-Min Chao. "Location Awareness in Ad-Hoc Wireless Mobile Networks." *Computer* 34.6 (2001): 46-52. IEEE Computer Society, June 2001. Web. Mar. 2011. <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=928621>.

Weisstein, Eric W. "Circle-Circle Intersection." *Wolfram MathWorld: The Web's Most Extensive Mathematics Resource*. MathWorld: A Wolfram Web Resource. Web. June 2011. <http://mathworld.wolfram.com/Circle-CircleIntersection.html>.

Xing, Guoliang, Chenyang Lu, Ying Zhang, Qingfeng Huang, and Robert Pless. "Minimum Power Configuration for Wireless Communication in Sensor Networks." *ACM Transactions on Sensor Networks (TOSN)* 3.2 (2007): 11-Es. *ACM Digital Library*. June 2007. Web. May 2011. <http://dl.acm.org/citation.cfm?id=1240231>.

Yang, Shuhui, Jie Wu, and Fei Dai. *Localized Movement-Assisted Sensor Deployment in Wireless Sensor Networks*. Tech. 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Oct. 2006. Web. July 2011. <http://www.cse.fau.edu/~jie/research/publications/Publication_files/TPDS-06-Scan-Based-Deployment.pdf>.

Zhang, Ying, Lukas D. Kuhn, and Markus P. J. Fromherz. *Improvements on Ant Routing for Sensor Networks*. Tech. Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence, Sept. 2004. Web. Feb. 2011. <http://www.lukas-kuhn.org/paper/nest-ant04.pdf>.