ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

COVERAGE ANALYSIS FOR 3-D REAL-WORLD TERRAIN MODELS WITH HETEROGENEOUS DEPLOYMENTS IN WIRELESS SENSOR NETWORKS

Ms. Thesis by

Anar KHALİLOV, B.Sc.

Department : Computer Engineering

Programme: Computer Engineering

DECEMBER 2007

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

COVERAGE ANALYSIS FOR 3-D REAL-WORLD TERRAIN MODELS WITH HETEROGENEOUS DEPLOYMENTS IN WIRELESS SENSOR NETWORKS

M.Sc. Thesis by
Anar KHALİLOV, B.Sc.
504051543

Date of submission: 13 December 2007
Date of defence examination: 26 December 2007

Supervisor (Chairman) : Doç.Dr. Sema OKTUĞ
Members of the Examining Committee Prof.Dr. Cem ERSOY (B.Ü.)
Prof.Dr. Muhittin GÖKMEN

DECEMBER 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ DUYARGA AĞLARINDA 3 BOYUTLU GERÇEK ARAZİ MODELLERİ İÇİN HETEROJEN DUYARGA KONUŞLANDIRMA STRATEJİLERİ İLE KAPSAMA ALANININ İNCELENMESİ

YÜKSEK LİSANS TEZİ

Müh. Anar KHALİLOV
504051543

Tezin Enstitüye Verildiği Tarih : 13 Aralık 2007
Tezin Savunulduğu Tarih : 26 Aralık 2007

Tez Danışmanı :	**Doç.Dr. Sema OKTUĞ**
Diğer Jüri Üyeleri	**Prof.Dr. Cem ERSOY (B.Ü.)**
	Prof.Dr. Muhittin GÖKMEN

ARALIK 2007

ACKNOWLEDGEMENTS

I am indebted to my mother Nushaba and father Chingiz who have been extremely understanding and supportive of my education throughout my life. And to my brother Elchin who always cared about me and my future.

I feel privileged to have worked with my supervisor Assoc. Prof. Dr. Sema Oktuğ. To her, I owe a great debt of gratitude for her patience, inspiration, friendship and support throughout my work. I would also like to thank her for giving me access to the Computer Networks Research Laboratory where I have been provided with an outstanding environment and friendly atmosphere during my research. It is certain that my work would not be what it is now without this kindness from her.

I also wish to thank Research Assistants who were constant source of answers to specific questions I had about anything during my efforts at the Laboratory.

I would also like to thank Prof. Dr. Engin Tarı and Assoc. Prof. Dr. Mustafa Yanalak for their guidance during the initial phase of the research.

I cannot ignore but mention my fiancée Merve Can Kuş for the unlimited patience and understanding shown by her at every point of my work. Her existence made my stay at the Istanbul Technical University very pleasant and it was her encouraging words that I heard firstly, it was her optimistic nature that made me approach problems calmly and dauntlessly.

Special thanks are due to my friends Rashad Gasimov and Shirali Shiraliyev who were always beside me when it came to encouragement and support. And I present my thanks to my cousin Farhad Hasanov for the interest and enthusiasm shown by him concerning my work. His ideas made me approach the problems from different perspectives and helped me to overcome the obstacles.

Anar Khalilov

December, 2007

TABLE OF CONTENTS

ABBREVIATIONS

API : Application Program Interface
DEM : Digital Elevation Model
DGM : Digital Ground Model
DHM : Digital Height Model
DT : Delaunay Triangulation
DTED : Digital Terrain Elevation Data
DTM : Digital Terrain Model
GLOBE : The Global Land One-km Base Elevation
LIDAR : Light Detection and Ranging
LoS : Line of Sight
SRTM : Shuttle Radar Topography Mission
TIN : Triangulated Irregular Network
VIP : Very Important Points
WSN : Wireless Sensor Network

TABLE LIST

Page No

FIGURE LIST

Page No

SYMBOL LIST

α	: Sensor output signal
p, q	: Arbitrary points
$d(p, q)$	: Euclidean distance between points 2 points, p and q
β	: Sensor specific constant
r_0	: Minimum distance threshold
c_i	: Initial coverage percentage at the peak
λ	: Elfes model constant
T	: Two-dimensional terrain
$\|T\|$	: Area of terrain T
$\|A_{pi}\|$	: Pseudo covered area of sub-triangle i, calculated using c_p
$\|A_i\|$	: Area of sub-triangle i
f_a	: Area coverage
f_n	: Node coverage
θ	: Minimum coverage threshold
c_p	: Coverage percentage
r	: Maximum sensing range of a sensor
S	: Set of sensors
s_i	: A sensor with index i
p_s	: Location of a sensor s
$O(p_s, q)$	: Sensor output signal for a sensor node as position p_s observing an event at position q
$I(q)$	: Sensor field intensity at point q

KABLOSUZ DUYARGA AĞLARINDA 3 BOYUTLU GERÇEK ARAZİ MODELLERİ İÇİN HETEROJEN DUYARGA KONUŞLANDIRMA STRATEJİLERİ İLE KAPSAMA ALANININ İNCELENMESİ

ÖZET

Kablosuz duyarga ağlarında kapsama alanı problemi önceki çalışmalara konu olmuş olsa da, bu çalışmaların hiçbiri kapsama alanını gerçek arazi modelleri üzerinde incelememiştir. Bu tezin amacı, "üç boyutlu gerçek bir arazi modelini belirli bir kapsama yüzdesi ile örtebilmek için kaç duyarga gerekli?" veya "artan duyarga sayısına uygun olarak kapsama alanı nasıl değişiyor?" gibi sorulara cevap bulmaktır. İlaveten, duyarga düğümlerinin yerleştirilme stratejisinin kapsama alanını veya kablosuz duyarga ağının enerji tüketimini nasıl etkileyebileceği incelenmektedir.

Gerçek hayattaki uygulamalara olabildiğince yakınsamak amacıyla, benzetimler gerçek arazi modelleri üzerinde çalıştırılmış, kapsama alanı gibi uygun büyüklükler hesaplanmış ve bunu gerçekleştirmek için birbirinden farklı pek çok benzetim senaryosu önerilmiş ve yürütülmüştür.

Önerilen çalışma kapsama alanı problemini bir daha gözden geçirip hangi parametrelerin kapsama alanı üzerinde etkili olduklarını bir daha ama bu sefer üç boyutlu ortamda incelemeyi amaçlamıştır. Elde edilen sonuçlar göstermektedir ki, uygulama gereksinimlerine bağlı olarak uygun duyarga modeli, uygun düğüm konuşlandırma stratejisi veya uygun duyarga sayısı seçilerek istenilen kapsama yüzdesi elde edilebilir.

Kablosuz duyarga ağlarında kapsama alanı problemi ilk kez bu çalışma sayesinde üç boyutlu uzayda gerçek arazi modelleri üzerinde incelenmektedir. Umulur ki bu çalışma yeni çalışmalar için temel oluşturur ve böylece önceden iki boyutlu uzayda incelenen tüm problemler şimdi yeni bir boyutun da eklenmesiyle üç boyutlu ortamda bir daha gözden geçirilerek daha gerçekçi sonuçların elde edilmesi böylece sağlanabilir.

COVERAGE ANALYSIS FOR 3-D REAL-WORLD TERRAIN MODELS WITH HETEROGENEOUS DEPLOYMENTS IN WIRELESS SENSOR NETWORKS

SUMMARY

Although the problem of coverage in wireless sensor networks has been studied in previous works, none of these suggestions analyzed coverage on real-world terrain models. The purpose of this thesis is to find answers to questions like how many sensors are needed to cover a specified 3-D terrain model at a specified coverage percentage. How actually coverage changes with increasing sensor count? Furthermore, how can a sensor deployment strategy affect coverage or energy consumption of a wireless sensor network? All of these questions are analyzed in the scope of this thesis.

A simulation environment was developed in order to analyze coverage on real-world terrain models from around the world to estimate the actual real-world entities as accurate as possible. In order to achieve this goal, numerous number of simulation scenarios were proposed and run.

The proposed work tries to summarize and make conclusions about area coverage and node coverage parameters and which input parameters affect them in three-dimensional space. Obtained results demonstrate that depending on application requirements an appropriate sensing model, or deployment strategy or sensor count can be selected so that the required coverage is obtained.

This is the first time that the coverage problem for wireless sensor networks is analyzed in three-dimensional space on real-world terrain models. Hopefully, this work can be used as a basis to build on. The problems previously considered in 2-D space can now be reanalyzed in a space with one more dimension, resulting in more accurate estimations.

1. INTRODUCTION

Wireless Sensor Networks (WSN) has become an extensively discussed topic in the last few years. A WSN is deployed over a region to sense events on geographical areas and transmit collected data to a sink node for further operations [1]. Depending on application requirements, it can be claimed that in most WSN applications it is very important to cover the region completely. In addition, in most WSN applications sensors are deployed over an outdoor environment. This may as well be a farm or a mountainous region; the point is that these regions have one thing in common: outdoor environment. In the literature, many algorithms are proposed and many implementations exist for the coverage issue. However, none of them address the coverage problem in a real-world scenario, over a real-world terrain model.

The goal of this thesis is to shed light over this subject by simulating sensor deployment on a real-world terrain model and make calculations about the coverage. Besides, the aim is to analyze how coverage changes by increasing sensor count, how node coverage (or in other words, node redundancy) is affected and etc. The main contribution of this thesis is that the simulation software is written for 3-D terrain models and has many customizable parameters like the selection of diverse deployment strategies, sensing models and etc. Additionally, this work is valid only for wireless sensor networks with sensor nodes that have optical or thermal sensors installed on them.

In the literature, there have been many studies but not one of them studied coverage over a real-world terrain model. Most of the simulations are conducted on a two-dimensional rectangular area with no obstacles at all. However, this is not the case in real-world scenarios.

In [2], Huang et al. discussed the coverage problem in three-dimensional wireless sensor networks where the goal was to determine whether a point is covered by at least k sensors where k is a given parameter. The same problem for 2-D was told to be solved in [3] and this work expands the solution to 3-D. A polynomial time

algorithm is proposed in this work. However, the coverage problem considered here deals with 3-D space without any obstacles and for a free space (e.g. questions like how many sensors are needed to cover the whole empty cube are addressed).

Another work similar to the one described in the previous paragraph was carried on by Watfa and Commuri [4]. Within the scope of this study, simulations were conducted to cover a region of 10x10x10 units with random sensor deployments. In this work, algorithms were proposed to select the minimum number of sensors required to cover such a region completely. “Coverage hole” problem is introduced and a substitute plan is suggested where sensor nodes that are almost out of energy are substituted with nodes to cover their region and conserve energy.

One more work described in [5] analyzes coverage mathematically from the probability point of view and a new algorithm – Configuration Algorithm based on Probability for Wireless Sensors (CCAP) is suggested. The simulations were run on a two-dimensional space and results of this algorithm are discussed.

Wan and Yi proposed another work [6] where sensors are deployed as a Poisson point process or a uniform point process on a square or disk region. Thereafter, the probability of k-coverage (target is being covered by at least k sensors) is analyzed accordingly to the changing sensing radius and the number of sensors. This work seems to contribute to the coverage problem significantly; however no 3-D scenarios are implemented in this study. Another work that concerns this thesis is proposed in [7] and it deals with maximizing the coverage of a given area with obstacles by introducing a centralized sensor deployment strategy. Another work for maximizing coverage is introduced by Rahman et al. in [8] where this is possible by determining least covered regions so that additional deployment is carried on later. The simulations are implemented in 2-D space.

To sum up, most of the work in the literature is focused on sensor placement (deployment) strategy or optimizing some parameters like the sensor count, network lifetime and etc. On the contrary, this thesis concerns more with how coverage changes according to relief changes. The answers for questions like how sensor deployment strategy affects node redundancy or how sensor count affects area coverage are investigated.

The thesis is organized as follows.

In Section 2, diverse sensing models are introduced. These sensing models are used as a part of our simulations. In other words, the simulations are carried out for different sensing models.

Terrain modeling which is discussed in detail in Section 3 is as much important as sensor modeling. After a brief introduction to digital terrain modeling, steps in terrain modeling are discussed. Thereafter, three different data models that are used in modeling a terrain are extensively discussed. Next, the methods that are used to create the models are discussed. And finally, coverage concepts like area coverage or node coverage are defined and explained in more detail.

Section 4 includes information about the simulation environment and input/output interface of the simulation. In other words, deployment strategies, sensing models, terrain models and sensor count parameters are explained in more detail and brief information is given about the implementation of the simulation. Moreover, the simulation results are discussed and plotted for visual assistance.

Finally, conclusions and future work is discussed in Section 5.

2. SENSOR MODELING

A sensor node is an integral part of a wireless sensor network. The main task of a sensor node is to collect data about the phenomenon being observed, make calculations if needed and transmit collected data to a sink node or to a gateway sensor node (other nodes may as well operate as relay nodes during this process). Main components of a sensor node are microcontroller, transceiver, external memory, power source and finally sensor(s) for observing environment [9].

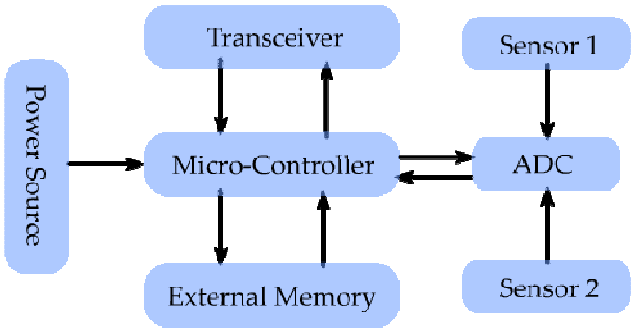


Figure 2.1. Sensor Node Architecture [9]

Sensor nodes, also known as motes vary in shape and physical dimensions depending on application requirements. Three different sensor nodes are demonstrated as an example in Figure 2.2 [10, 11, 12].

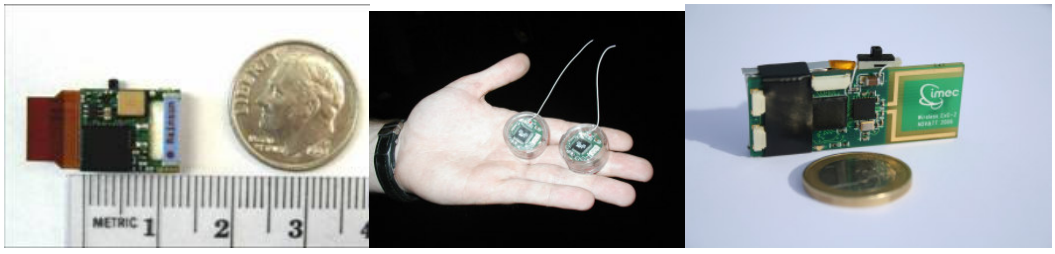


Figure 2.2. Wireless Sensor Nodes [10, 11, 12]

Sensors assembled onto a mote may also be of a varying shape and size. There are sensors for measuring physical entities such as pressure, temperature, light. In other words, a sensor transforms a real-world entity into an electrical signal. The quality (i.e. accuracy) of the signal depends on several factors [13]. One of those factors is

the distance between the sensor and the point of observation. Some signals fade according to the power law (e.g. acoustic signals fade quadratically). Another factor that affects the sensor output signal quality is directionality, i.e. not all directions are equally sensed in practice. This may be due to hardware imperfections or partial breakdowns of mote components as a result of coarse deployment. The third and the last factor is the instability of the measured values. In other words, under exactly the same circumstances, a sensor may output close, yet different values.

When considered in practice, these factors are generally subject to several assumptions. In most cases, sensors are assumed to be omnidirectional which means that they are capable of sensing in all directions equally. It is also assumed that the measured values are stable under exactly the same circumstances. Only the first factor is generally taken into account when dealing with wireless sensor networks in practice.

2.1 Sensing Models

Considering these conditions, sensing models can be classified into three as the binary sensing model, the exponential sensing model and the hybrid sensing model.

2.1.1 Binary Sensing Model

The model is quite straightforward: if the sensed phenomenon is within a pre-defined sensing radius *r*, then it is definitely sensed. Otherwise, the sensor is not capable to sense the object of interest at all. The model is also known as the Boolean sensing model. As proposed in [14], sensor output signal can be modeled as follows:

$$O(p_s, q) = \begin{cases} \alpha : d(p_s, q) \le r \\ \\ 0 : otherwise \end{cases} \quad \textbf{(2.1)}$$

where p_s denotes the position of the sensor and similarly q denotes the position of the target. Moreover, α is used to show the sensor output signal value and $d(p_s, q)$ is the Euclidean distance between two points – the sensor and the target.

In the thesis, α is assumed to be not an electrical entity (voltage), but rather the probability of a successfully sensed event. For the binary sensing model, for an

arbitrary point within the range of a sensor node, this probability is equal to 1 (i.e. event is reliably sensed) by the definition itself. Detection probability function for the binary sensing model is illustrated in Figure 2.3.

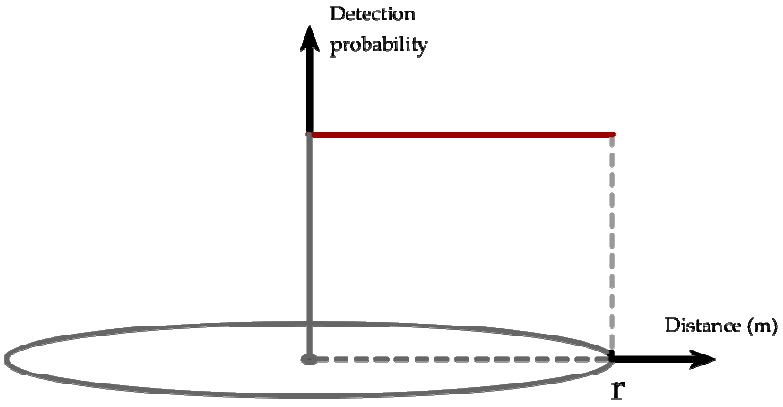


Figure 2.3. Binary Sensing Model

We may also refer to the probability of a successfully sensed event as coverage percentage. This concept will be explained in more detail later.

In [15] it was proposed that sensors may have varying radius or sensing range depending on their residual energies. However, in this thesis, this assertion is ignored.

2.1.2 Exponential Sensing Model

This is a more comprehensive model when compared to the previous one. Typically, in models where sensor output signal varies with distance, the exponential sensing model is more appropriate to use. Sensor output signal can be modeled as follows:

$$O(p_s,q)=\begin{cases}\dfrac{\alpha}{d(p_s,q)^{\beta}} : r_0 \le d(p_s,q) \le r \\ \\ 0 : otherwise\end{cases} \qquad \textbf{(2.2)}$$

where β is a positive real number depending on the sensor type. For instance, it was shown in [16] that acoustic signals can be modeled with $\beta = 2$. Similarly, magnetic sensors obey a power law with $\beta = 4$.

In Equation (2.2), r_0 denotes a minimum distance threshold in order to prevent division by zero. In this work, r_0 is chosen to be 1/10 times the sensing range and β is equal to 2. Again, α is sensor output signal value. Detection probability function for the exponential sensing model is demonstrated in Figure 2.4.

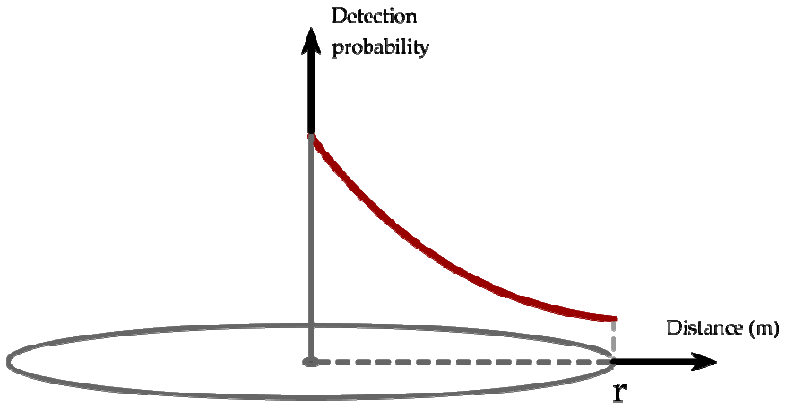


Figure 2.4. Exponential Sensing Model

In the literature, there have been other sensing models like Neyman-Pearson model [17, 18] which is very similar to the exponential sensing model from the concept point of view.

For the Neyman-Pearson model, the detection probability of a target at grid point v by sensor i is:

$$p_{vi} = 1 - \Phi(\Phi^{-1}(1-\alpha) - \sqrt{\gamma L d_{vi}^{-\eta}}) \quad \textbf{(2.3)}$$

where d_{vi} is the Euclidean distance from point v to point I, α is false alarm rate, γ is signal-to-noise ratio, L is the number of data samples, Φ is the cumulative distribution function of the zero-mean, unit variance Gaussian random variable at point x. For more detail, the reader can refer to [17].

2.1.3 Hybrid Sensing Model

Another model was firstly introduced by Elfes as described in [19]. Here, the probability of sensing is a function of 2 parameters: λ and β. These parameters are used to model different sensor characteristics:

$$p_{vi} = \begin{cases} 1 : r - r_e \geq d_{vi}, \\ e^{-\lambda(d_{vi} - r + r_e)\beta} : r_e > |r - d_{vi}|, \\ 0 : d_{vi} \geq r + r_e \end{cases} \quad \textbf{(2.4)}$$

where r_e is a measure of uncertainty in sensor detection. Here, the function is constant until some point and then it decreases exponentially.

Being inspired by this model, we propose a similar but a simplified model.

The purpose is to combine previously presented two models. It would be interesting to analyze such a hybrid model in order to observe the differences between the hybrid sensing model and previously presented two models. In other words, to see the changes more smoothly such an intermediate model will also be simulated. The model is described in Figure 2.5.

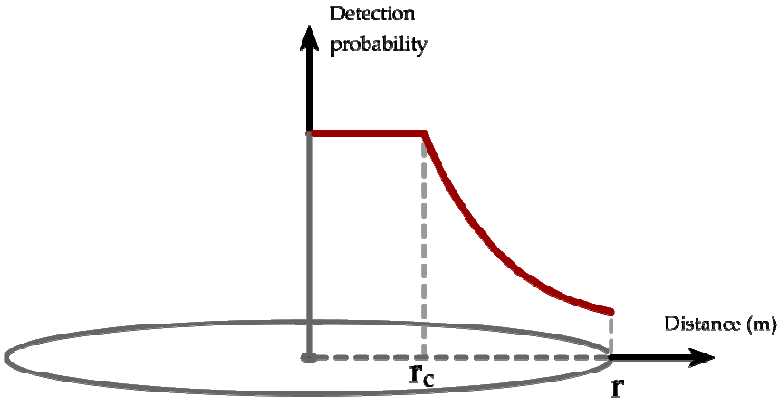


Figure 2.5. Hybrid Sensing Model

Such a model can be defined using the following formula:

$$O(p_s, q) = \begin{cases} 1 : r_c \geq d(p_s, q) \geq 0 \\ \dfrac{\alpha}{d(p_s, q)^{\beta}} : r \geq d(p_s, q) > r_c \\ 0 : otherwise \end{cases} \quad \textbf{(2.5)}$$

where r_c is a predefined threshold until which the value of the function is constant and after that point the value decreases exponentially.

2.2 Sensor Characteristics

In Figure 2.1 main components of a sensor node are given. It is shown in the figure that a sensor node may have more than one sensor installed on the mote, connected to the Analog-Digital Converter (ADC) component. However, in the scope of this thesis, it is assumed that only one type of sensor is installed on a sensor node in the network. In practice, there exist different types of sensors like magnetic sensors, thermal sensors, chemical sensors, and etc. For a detailed analysis of these sensors the reader can refer to [20].

Since sensors have different characteristic properties, it is expected that their performances will also be different than each other. To give an example, under exactly the same circumstances magnetic sensors will produce different results than acoustic sensors as explained in Section 2.1.2. In this scope, it is important to determine which sensor types are suitable for this thesis.

In this thesis, it was assumed that sensors can communicate with each other only if there is a line-of-sight between sensor nodes. Signal reflection or scattering was not considered in the scope of the thesis. Additionally, the sensing range of a sensor node was chosen to be 25 meters.

Under these two circumstances, many sensor types are not feasible for the work presented in this thesis. For example, magnetic sensors cannot be used because of their short range [20, 21]. Radar is not appropriate sensor type for this work because it is known that radar can operate even when line-of-sight does not exist through the reflected signals [22]. Similarly, the same can be claimed for acoustic and seismic sensors.

Considering these assumptions, it was decided that the results of this work are valid only for sensor nodes equipped either with thermal or optical sensors. Thermal sensors can operate for long distances through their infrared cameras. Thermal sensors can be used in target detection applications. It is a fact that human body emits heat as infrared energy. The similar can be said about hot spots of the vehicles like engines. Considering these facts, thermal sensors can be used in target detection

applications to detect animals, human or vehicles [23]. Optical sensors operate in the similar manner [24] making this type of sensors also suitable for this thesis.

3. DIGITAL TERRAIN MODELING

In this section, technical background about digital terrain modeling is presented. Firstly, a brief introduction defining the concept of a digital terrain model is given. Thereafter the data models that can be used to model a real-world terrain are presented and also public elevation data sources are given. And finally, the concept of coverage for wireless sensor networks is explained in detail.

3.1 Introduction to Digital Terrain Modeling

During the late 1950s two American engineers at the Massachusetts Institute of Technology introduced a new term – digital terrain model (DTM) which was meant to create digital models of real-world terrains. The explanation of this new term - DTM was as follows: "DTM is simply a statistical representation of the continuous surface of the ground by a large number of selected points with known X, Y, Z coordinates in an arbitrary coordinate field" [25]. As time passed by, new terms such as digital elevation model (DEM), digital height model (DHM), digital ground model (DGM), and digital terrain elevation data (DTED) were defined to describe the similar concepts. These terms are often thought to be the same in practice, but actually they refer to distinct concepts [26].

DEM and DTM are the most extensively used two terms in practice. DTM often refers to the altitude of the ground itself whereas DEM refers to the maximum altitude everywhere (including buildings, trees, etc.). Thus, digitizing topographic maps will yield a DTM, while deriving data from satellite imagery will supply DEMs.

As it is implemented in most applications, a 3-D image model of a terrain may as well be generated using a DTM and a satellite image corresponding to that terrain. This will produce a much more clear idea of the terrain being analyzed in terms of its visual interpretation. A sample 3-D image model is demonstrated in Figure 3.1.

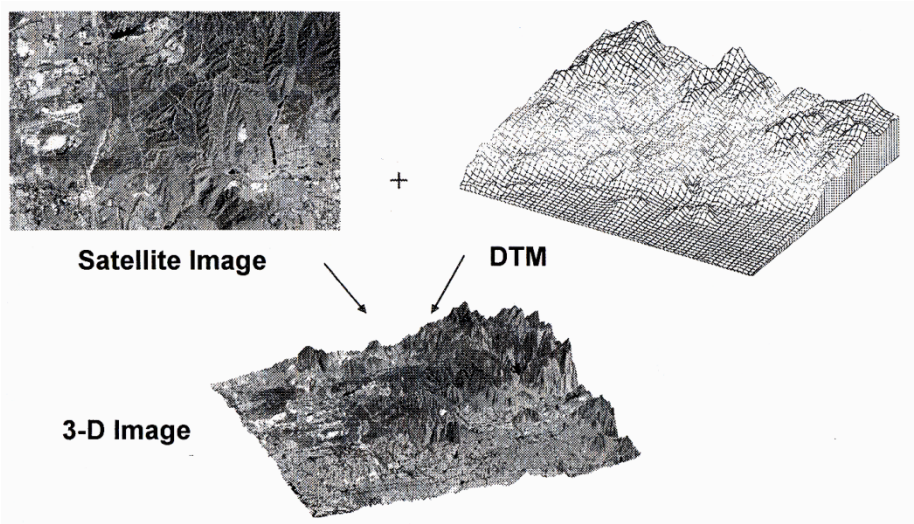


Figure 3.1. Satellite imagery [27]

3.2 Steps in Digital Terrain Modeling

Digital terrain modeling includes five separate operations: generation, manipulation, interpretation, visualization and application [27].

3.2.1 DTM Generation

This process is very critical in terms of accuracy and cost since subsequent steps are directly affected by the outcomes of this process. There are several choices to generate a DTM and the selected method will always be "a balance between the desired accuracy of the DEM and the costs involved in its creation". When it comes to selection, there are free, low accuracy products (e.g. GTOPO30); more costly, medium accuracy products (satellite data) and high accuracy products extracted from airborne sources like LIDAR using photogrammetry.

There are typically two approaches when representing DTMs: an image method (point or line model) and a mathematical method. There have been extensive research and tests with numerous data structures to represent a terrain digitally but two models are of special importance because of their simplicity and popularity. The first one models terrains using rectangular elevation matrices, whereas the second one uses another approach where terrain is represented using neighboring triangles i.e. triangulated irregular network (TIN). However, both of these representations have one thing in common: they both use the point model. The rectangular grid matrix is the most extensively used representation because of its simplicity. The

model can be stored in a straightforward data structure used in modern computer architectures – a two-dimensional array. The TIN model in its turn has own advantages like its ability to reserve more data space where changes in relief are very much and less space to regions where changes are less, thus minimizing the data redundancy. Both of these representation models will be discussed in further chapters.

3.2.2 DTM Manipulation

This process includes changing, adjusting, refining the DTM which was obtained in the first step of generation. Further operations such as editing, filtering, merging and joining are also accomplished at this step. The purpose of editing is correcting errors existing in the DTM. Filtering is required to smooth (reduce details) or enhance (emphasize details) the terrain using low-pass or high-pass filters, respectively. Finally, merging and joining steps are used to combine more than one DTM.

3.2.3 DTM Interpretation

The purpose of this process is to analyze the given DTM and make corresponding conclusions. Roughly speaking, the interpretation of a DTM can be in the form of providing information about the slope, gradient and etc. The process of interpretation is especially important from the geo-morphological modeling point of view.

3.2.4 DTM Visualization

DTM Visualization serves as an important process of visually interpreting the digital terrain model and has a significant importance in terms of perceptual understanding of the DEM. Numerous tools can be obtained through the World Wide Web for the visualization purposes.

3.2.5 DTM Application

There are a large number of applications for DTMs in the fields such as military, environment, engineering and commerce. Main application domains of DTMs include civil engineering, Earth sciences, planning and resource management, remote sensing, military applications and etc.

3.3 Data Models in DTM

When obtaining the terrain data, the collected information is usually not very well arranged; i.e. it needs to be bound to a model that will store all related information about the terrain. Since the Earth has a continuous surface and since any part of this surface can be represented using infinite number of discrete points, a sampling method needs to be defined in order to be able to construct the surface model. A surface model must satisfy the following properties [28]:

1. Accurately represent the surface
2. Be suitable for efficient data collection
3. Minimize data storage requirements
4. Maximize data handling efficiency
5. Be suitable for surface analysis

There are mainly three models that are used for this purpose (i.e. to digitally model a surface): contours, grids (lattice or elevation matrix) and triangulated irregular networks (TIN).

3.3.1 Contours

Contour maps are the main source and a self-explanatory example of this model. Since contour map of the whole world is available, this model is deemed to be the most common one. However, it has its own undesirable properties. Contours are formed using the isolines which connect points of the same height with continuous lines; practically curves. Under these circumstances, elevation data between these lines is not available and they must be obtained using an interpolation method. In other words, only points on isolines carry the real elevation data; all other points are interpolated when needed.

The quality of contours depends on the method they were obtained. When obtained from aerial photographs, the values are quite accurate. On the other hand, when point data is obtained using an interpolation method, the accuracy might suffer dramatically.

3.3.2 Grids

Grid model is probably the most widely used surface model due to its simplicity. There is a close similarity between an elevation matrix and a two-dimensional array structure of digital computers. Physical dimensions of a surface will directly define the dimensions of a two-dimensional elevation matrix. The distance between consequent elements of the array will therefore represent the sampling interval of the terrain which is in fact constant for all successive points. Since the array data structure corresponds to the surface model, it is known that the algorithms for this model will also be quite straightforward. This is probably the biggest advantage of this structure. As depicted in Figure 3.2, a sample grid model can be in the form of a two-dimensional matrix.

$$\begin{bmatrix} 23 & 24 & 24 & 18 \\ 22 & 23 & 23 & 19 \\ 22 & 23 & 23 & 20 \\ 21 & 23 & 23 & 20 \\ 21 & 24 & 23 & 19 \end{bmatrix}$$

Figure 3.2. A grid model data structure

A value in this matrix denotes the amount of elevation which was sampled when extracting this grid from a real-world terrain. If let say these values were sampled at every 50 meters, then this (5 × 4) grid matrix would represent a terrain with dimensions of 200 × 150 meters. Since the surface is divided into cells (unit squares) of 50 × 50 meters, it is obvious that the amount of elevation at every point inside a cell is assumed to be equal. And this is quite an enormous drawback of this model.

Moreover, the grid model can be formulated using two different ways: lattice or categorical approach. In the latter one, the points have the same elevation amount at any given cell of the array; i.e. the elevation is thought to be as a flat plane at the given point. However, in the lattice structure, the mid points of neighboring cells are evaluated and then connected. This results in differentiated height values through the region. Figure 3.3 is given to present visual understanding of these two different approaches.

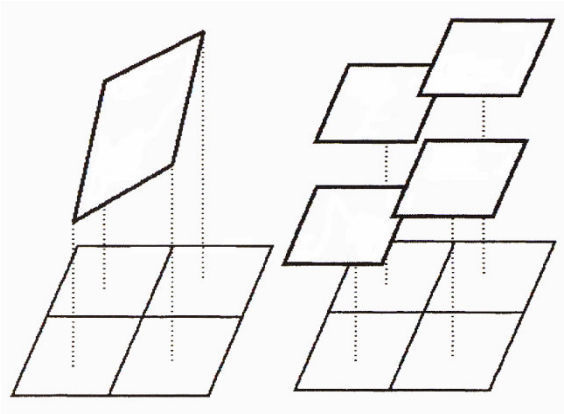


Figure 3.3. Lattice versus categorical approach [27]

Obviously, the accuracy of the grid model directly depends on the pre-selected sampling interval. When chosen to be adequately large, more points will be ignored. However, when the sampling interval is small, this time accuracy requirements will be met at cost of data storage requirements. This trade-off can appropriately be adjusted depending on application requirements and characteristics of the relief. Furthermore, it should be kept in mind that under sampling and over sampling will only affect the surface model negatively in terms of accuracy and data storage requirements, respectively.

3.3.3 Triangulated Irregular Network (TIN)

The triangulated irregular network model was firstly introduced and then developed by Peucker et al. in the early 1970s and nowadays is a valuable alternative to the previously explained grid model. The main idea behind this model is that in this model, a surface is modeled using the irregularly distributed points in three-dimensional space using a set of triangles.

Obviously, the biggest benefit of this model is that some areas with high details can be modeled using more points whereas few points are needed to represent smooth areas. To recall, in the grid model, all areas are represented by the same number of points (i.e. regularly distributed points) which naturally results in loss of detail or in data redundancy. As a result, it is certain that the TIN model represents the surfaces more efficiently. In addition, the TIN model can estimate the real-world terrain more accurately (i.e. include details), when compared to the gird model, however the algorithms needed to process the TIN model are more complex than that of the grid model.

To explain TIN further, let us observe its elements. A triangulated irregular network mainly consists of nodes and edges that form main component of a TIN – triangles. Nodes are nothing but elevation points whereas edges are line segments that connect these points to form triangles. Every point has 3 Euclidean space coordinates: x, y and z. Edges connect these points (or nodes) in such a way that they form triangles that satisfy the Delaunay triangulation condition. Delaunay triangulation will be explained later.

The table of corresponding advantages and disadvantages of Grid model and TIN model is demonstrated in Table 3.1 [29].

Table 3.1. Comparison of Grid model and TIN model [29]

	Grid DTM	*Triangulation*
Smoothing	Due to the generation algorithm based on least squares adjustment, grid-based methods perform smoothing.	Difficult to achieve because the original data points are used.
Geomorphology	Break lines can be considered.	Break lines can be considered.
Point density	Fixed due to the matrix structure. Usually, a trade-off between storage capacities and precision is sought.	Variable as the original data points are used.
Robustness	Robust estimation procedures can be applied. Problems appear with inhomogeneous point distributions.	Problems due to the nonuniqueness of the ordering criterion.
Applicability	Restrictions due to the 2.5-D characteristics. Simple algorithms can be found for many tasks.	More general than grid-based methods but also restricted. More difficult algorithms are required.

In this thesis, the triangulated irregular network model was preferred due to its accuracy and all other previously stated advantages.

3.4 Elevation Data Sources

There exist many global elevation data sources and most of them have been developed over the last decade [27]. These data sources are introduced briefly as follows.

3.4.1. The Global Land One-km Base Elevation (GLOBE)

This work was initiated by Committee on Earth Observation Satellites (CEOS) and has a grid spacing of 1 km [30]. This data set covers 180° west to 180° east longitudes and 90° north to 90° south latitudes.

3.4.2. Digital Terrain Elevation Database (DTED)

Developed by National Imagery and Mapping Agency (NIMA), this model has a horizontal grid spacing of 30 arc-seconds which is approximately 1 km [31]. There are three levels of this data model available, but only Level 0 is available to public. Level 1 and Level 2 are of a higher resolution and accuracy, but these data sets are not available to public.

3.4.3 GTOPO30

This global elevation model has a horizontal grid spacing of 30 arc-seconds which means that points are sampled at every 900 meters [32]. This data set was created by U.S. Geological Survey's EROS Data Center in three years.

3.4.4 Shuttle Radar Topology Mission (SRTM)

This work was initiated by National Aeronautics and Space Administration (NASA) and National Geospatial-Intelligence Agency (NGA) [33]. The difference of this project from the previous ones was that in SRTM cutting edge technology was used so that weather conditions or cloudiness did not affect measurements that much. There is a data set named as SRTM1 with horizontal spacing of 1 arc-second (approximately 30 meters), but it is only available for North America. The data set that includes the whole Earth is only available with horizontal spacing of 3 arc-seconds (90 meters) and this level is called SRTM3. There exists also the SRTM30 level (approximately 1 km of grid-spacing), but since it is not accurate enough, this level is ignored in the scope of this thesis.

Since the previously introduced three elevation data sources are not accurate enough, the models in this thesis will not be formed using these data sets. The elevation data used in this thesis was taken from SRTM ftp site [34]. Some specific areas around the World were chosen for simulations. The available data is in the .hgt file format and in order to process these files, a tool named 3DEM [35] was used. Through this

terrain visualization tool, the topography of a terrain model can easily be observed. The SRTM data file can be converted to a "Terrain Matrix" consisting of signed integers and can be saved as a binary file. These binary files include subsequent height values of the terrain. By knowing the width and length of the terrain, the files can be easily manipulated and be read to a two-dimensional array for further processing. But since SRTM provides elevation data in the Grid model format, firstly the data pertaining to this model must be converted to an appropriate data format for the TIN model.

In the thesis, the SRTM3 data was used to create terrain models. However, it was decided that the resolution of 90 meters was not accurate enough to observe and make conclusions about coverage. Eventually the SRTM3 data was used but it was interpreted as if it was a grid with sampling interval of 5 meters (i.e. the scale was modified from 90 meters to 5).

3.5 Conversion from Grid Model to TIN Model

The process of converting from one model to another can be divided into two individual independent sub-problems: picking significant points and connecting these points appropriately to form triangles.

3.5.1 Significant Point Selection

In order to implement this task, there exist several algorithms; namely Fowler and Little algorithm [36] and Very Important Points (VIP) [37]. In this thesis, the latter algorithm was used when converting SRTM grid data to create a TIN model of the same surface. Before starting to explain these algorithms, it is necessary to explain the concept of significant points.

Significant points are the points that mainly define the surface. In other words, these are the points where sharp variations in relief occur (i.e. slope at these points is discontinuous). With the removal of these points, a significant loss in terms of details will occur. Similarly, points that do not contribute to the relief shape variations that much are said to be "not very significant" points. In input grid matrix, for each value (cell) this significance value is calculated using its neighboring cell values and then depending on the application requirements, some of these points are eliminated,

leaving only the most significant points. Significant points are also called as Very Important Points.

As it can be predicted, points on smooth areas will have significance values closer to zero, whereas points where sharp changes occur will have larger values and such a point will be called a VIP. Figure 3.4 demonstrates points with diverse significance values.

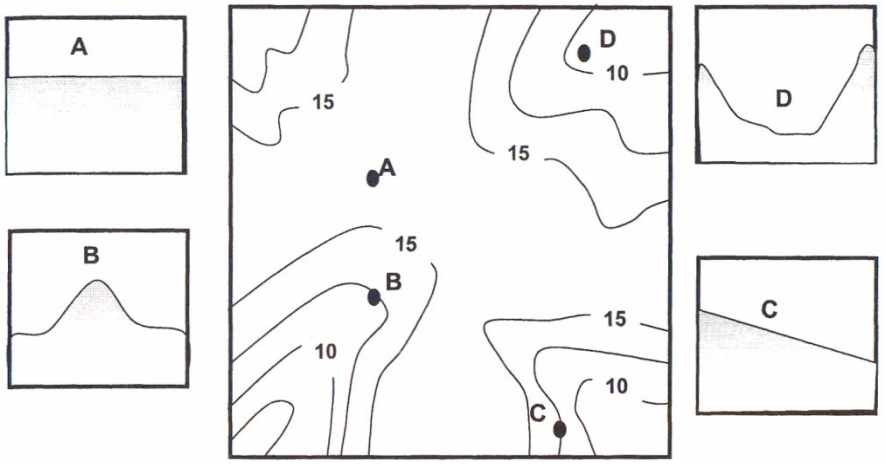


Figure 3.4. Very Important Points [27]

In Figure 3.4, it can easily be predicted that points A and C are not as significant as points B and D because of the sharp relief changes. By eliminating A or C, not much detail will be lost. However, the same cannot be claimed about points B or D.

The "Very Important Points" algorithm was proposed by Chen and Guevara in 1987. Initially, a terrain is in the form of a two-dimensional elevation matrix and naturally, values that are not at the borders of this matrix will have 8 neighbors and 4 diametrically opposite pairs. These four pairs are up and down, right and left, upper-left and lower-right and finally upper-right and lower-left. For each of these pairs a closeness degree is calculated [38]. These parameters define the amount of closeness of a point inside a terrain to its neighborhood. Eventually, four values are obtained since there are four diametrically opposite pairs and the average of these values will result in a significance value of the point under consideration.

The elimination step of not very significant points can be executed by simply sorting these significance values and filtering only the ones that exceed some pre-defined threshold value.

Each of the terrain models used in this thesis has 201 rows and 201 columns which makes a total of 40401 points. 30401 of them are eliminated and 10000 are left for further steps.

3.5.2 Delaunay Triangulation

When only the very important points are left, the TIN model has to be constructed from these points and to accomplish this task, generally the Delaunay triangulation method is used due to its robust characteristics. The most important property of Delaunay triangulation is that it maximizes the minimum angle and avoids narrow triangles. This is a significant property since the triangles produced by this triangulation method will be as similar as possible to an equilateral triangle.

The purpose is to connect sample input points using line segments by forming non-overlapping triangles in such a way that no sample point is inside the circum-circle of any triangle. The circum-circle of a triangle is a circle which passes through all of the vertices of the triangle. The center of a circum-circle is called circum-center. A sample triangle with its circum-circle is given in Figure 3.5.

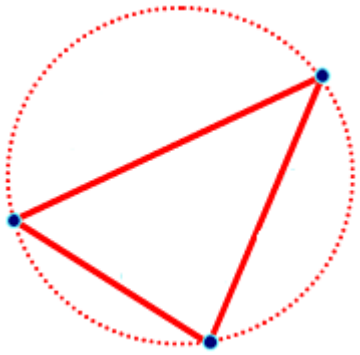


Figure 3.5. A triangle and its circum-circle

In most cases, the Delaunay triangulation is unique. For example, if points lie on four corners of a rectangle, then triangulation may be carried on in one of two ways. However, these types of conditions occur rarely in real-world problems. A simple Delaunay triangulation of 100 points in two-dimensional space is depicted in Figure 3.6 [39].

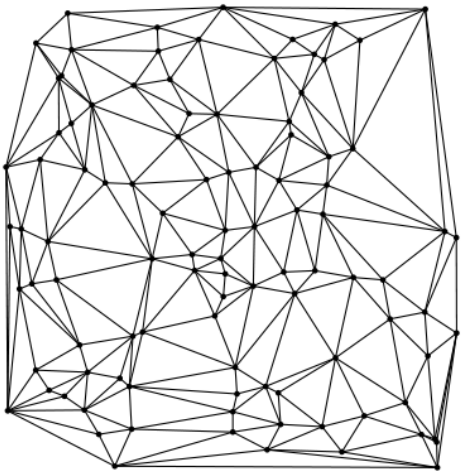


Figure 3.6. A simple Delaunay triangulation [39]

Additionally, it can be claimed that the most important application of Delaunay triangulation is terrain modeling. When triangulating a terrain, regions with more details will generate more triangles than the regions with fewer details. This is a very significant issue because in some applications the model may be required to estimate the real-world phenomenon as much as possible.

In 1989, Bourke [40] developed an efficient triangulation algorithm that is suitable for terrain modeling and in this thesis his algorithm was used when obtaining Delaunay triangulations. The algorithm is dynamic, i.e. when more points are added to the model the triangulated model can easily be updated. This makes it possible to construct the triangulated model from scratch. In other words, one can construct a TIN having a set of points and by adding one point at each step and this triangulation will be unique.

3.6 Dividing TIN into sub-triangles

Coverage measurements can be easily calculated in grid-based surfaces because the terrain is divided into equal portions and in order to find the coverage percentage, it is enough to divide the area covered by sensors to the total area of the surface. However, in triangulated irregular networks, it will not be an easy task to fulfill since the triangles will have different size and shape. One solution can be proposed by dividing triangles of a TIN into smaller sub-triangles. When these newly created sub-triangles are sufficiently small, then it will be much easier to decide whether a sub-

triangle is covered or not. By this division, it looks like as if the TIN became a grid and consequently, coverage calculations become less complicated.

The process of dividing triangles (i.e. elements of a TIN) into sub-triangles can be carried out using several techniques that are explained further.

3.6.1 Subdividing a Geodesic Face

Each side of the triangle is subdivided into smaller parts. In other words, n points are defined on each edge of the triangle in such a way that the distances between these points are equal. Next, a total of $2 \times n$ points on two sides of this triangle are connected using n lines as described in Figure 3.7a.

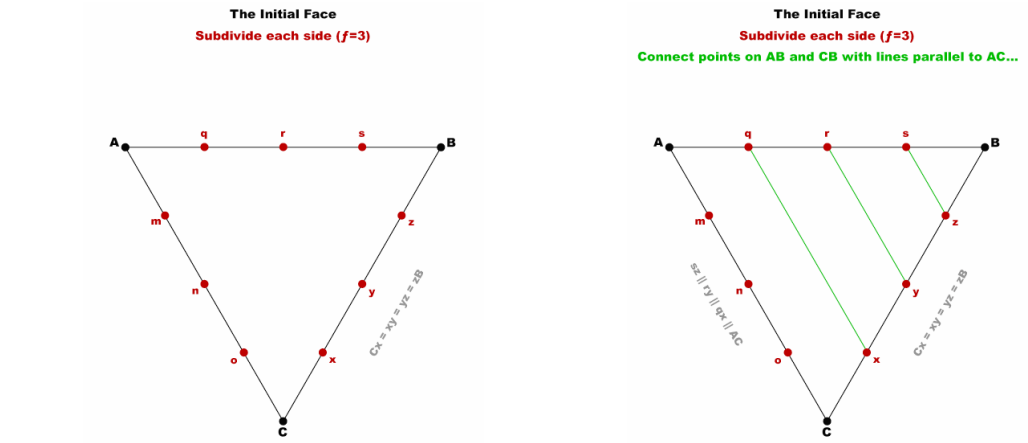


Figure 3.7a. Subdividing a geodesic face, step 1 and 2 [41]

Step 2 is applied to all pairs of triangle sides. New points are actually the intersection points and new sub-triangles can easily be formed using this technique as described in Figure 3.7b.

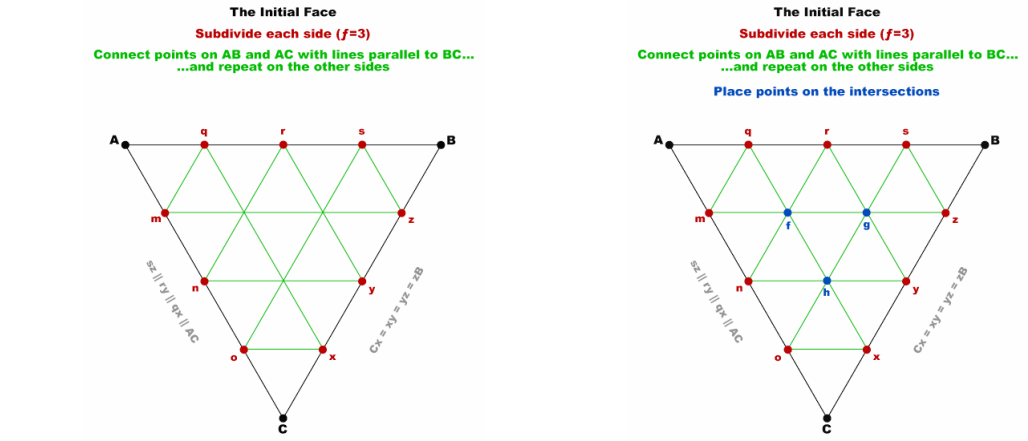


Figure 3.7b. Subdividing a geodesic face, step 3 and 4 [41]

In the end, the triangle will be divided into smaller sub-triangles using this simple technique. The size of newly created sub-triangles can be controlled using *n*.

3.6.2 Simple Median Partitioning

When compared to the previous technique, the one presented in [42] seems to be simpler from the complexity point of view. The idea is simple; a triangle can be divided into two sub-triangles by drawing a median to its largest edge. Then, newly generated sub-triangles can be divided in the similar manner as depicted in Figure 3.8 and this process can last forever.

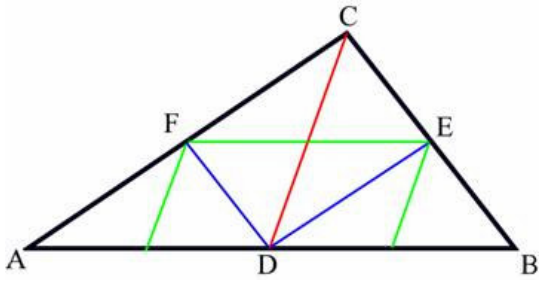


Figure 3.8. Simple Median Partitioning [42]

At first, the triangle had only 3 points: A, B and C. The red line segment CD was used to divide it into two sub-triangles: ADC and CDB. Then blue lines can be used to similarly to generate sub-triangles ADF, DFC, DCE and DEB. The operations can carry on until required conditions are met.

In this thesis, the "Simple Median Partitioning" technique was used to divide elements of the TIN – the triangles. The division operation is carried on until the newly created sub-triangles are small enough. This is checked by evaluating the perimeter of triangles. A perimeter threshold is defined and compared to the perimeter of a sub-triangle. If perimeter is greater than the threshold, then the sub-triangle is divided. In all other cases, the sub-triangle is said to be small enough and no other operation is conducted. The sub-division process is carried on recursively.

3.7 Coverage

In many wireless sensor network (WSN) applications, the main task is the surveillance of certain geographical areas. There is a wide range of applications in wireless sensor networks for detecting intruders, preserving rare species, preventing

fire and etc. In all of these applications, there is a need to cover the area of interest in order to successfully sense nearby events.

In wireless sensor networks, it is known that besides covering an area, it is also important to exploit the sensor density factor in such a way that some sensors can be switched into sleep mode without reducing the coverage. In such a way, network lifetime can be significantly improved.

3.7.1 Types of Coverage

Typically, coverage deals with a two-dimensional terrain T, having an area of $|T|$. Two types of coverage are defined in the literature [14].

3.7.1.1 Area Coverage

The area coverage of terrain T is defined as the percentage of $|T|$ being covered and is specified by f_a. Since it is a percentage entity, f_a is defined to be between 0 and 1 (inclusive at both sides). When f_a is 0, there is no coverage of terrain T at all. Similarly, when f_a is equal to 1, it can be concluded that full area coverage is attained.

Consider an arbitrary point $q \in T$ which is to be covered. $I(q)$ is defined as sensor field intensity [43] at the point q and is calculated using Equation 3.1.

$$I(q) = \sum_{s \in S} O(p_s, q) \tag{3.1}$$

where S is the set of sensors and p_s is the location of a sensor s. The function $O(p_s, q)$ is defined as the sensor output signal for a sensor node at position p_s observing an event at position q. In other words, $I(q)$ is the total sensor output signal at point q. Depending on applications, $I(q)$ could be a measure of coverage. By defining some coverage threshold θ, a point q is said to be covered if and only if $I(q)$ is greater than or equal to θ. When a point is covered by more than one sensor (i.e. $k > 1$ sensors), it is referred as k-coverage problem [44].

3.7.1.2 Node Coverage

Node coverage f_n is defined as percentage of nodes whose sensing range can be completely covered by the sensing ranges of other sensor nodes. This concept makes

sense especially in wireless sensor network applications because since sensor nodes take their energy from batteries, the conserved energy becomes an invaluable factor from the network lifetime point of view. To explain further, nodes whose sensing range is fully covered by the sensing ranges of other nodes can be safely switched into the sleep mode without reducing area coverage.

When compared to area coverage, node coverage generally takes smaller values. In order to improve it, the only solution is to increase sensor density (i.e. number of sensors deployed over a unit area). Figure 3.9 demonstrates the relationship between area coverage and node coverage [14].

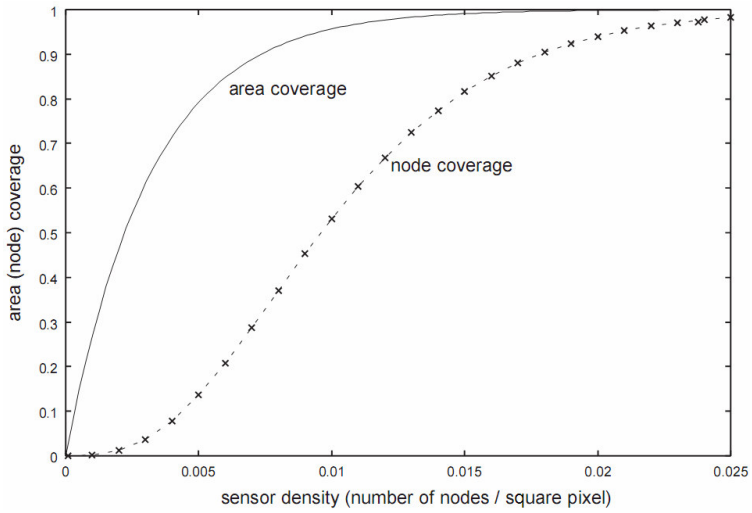


Figure 3.9. Area coverage vs. node coverage [14]

It is evident that area coverage increases faster than its other counterpart - node coverage. To better see the idea, consider the following relationship between these two parameters: in order to satisfy a 50% node coverage requirement, the area coverage must be approximately 95%. Similarly, when node coverage is near 1%, area coverage of approximately 50% has already been reached. Thus, it is not sufficient to improve area coverage parameter only; but to also improve node coverage if network lifetime requirements are taken into account.

In order to see the relationship between these two parameters more clearly, the work by Liu and Towsley [45] was analyzed. In this work, simulations were conducted on a two-dimensional space of 10000 × 10000 pixels. The node density parameter represents the number of nodes deployed over a unit area and is measured in pixels^{-2}. Each node has a sensing radius of 10 pixels and the locations of sensors are described

by a Poisson point process. Figure 3.10 demonstrates how area coverage can be different than node coverage:

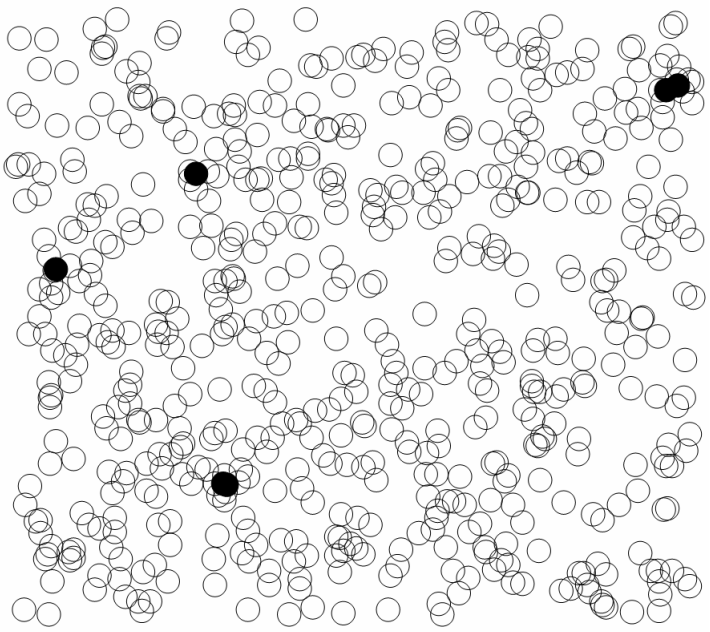


Figure 3.10. Part of the network at node density 2×10^{-3} [45]

Here, nodes filled with black color represent the nodes whose sensing areas are completely covered by the sensing range of other nodes, i.e. these nodes are redundant nodes. Only a tiny fraction of all nodes are redundant. To improve this, the only solution is to increase sensor density either by decreasing the area or by increasing the number of sensor nodes.

In this thesis, area coverage parameter was calculated by dividing the covered area (summation of all covered sub-triangles) to the total area of the terrain. Similarly, node coverage was calculated by dividing the number of redundant nodes to the total number of sensor nodes. A more detailed explanation is given in Section 4.

3.7.2 Coverage Implementation

It was previously stated that the whole relief was modeled as a TIN model. To make things simple, the elements of the TIN – i.e. triangles were divided into sub-triangles as explained in Section 3.6. Therefore, coverage of a sub-triangle (not a particular point) was taken into account. Since 3 different sensing models were considered in the thesis, it is necessary to consider the re-definition of coverage for a sub-triangle for each of these models.

A sub-triangle is said to be covered (fully or partially; depending on the sensing model) by a sensor node, if all of the three points forming the sub-triangle are inside the sensing range of the sensor node and there is a line-of-sight between the sensor node and all three vertices of the sub-triangle (see 3.6.2.2).

3.7.2.1 Coverage Percentage

For the binary sensing model, if a sub-triangle is covered by a sensor node, it is said to be fully covered (equal to 100%). However, in models where sensor output signal is a function of distance between the sensor itself and the observed target, a sub-triangle is said to be partially covered.

At this step and in this context, it would be appropriate to introduce a new term – coverage percentage. In the scope of this thesis, this term is equivalent to the probability of a successfully sensed event. This definition becomes useful when dealing with the exponential sensing model where the probability of a successfully sensed event changes according to the sensor-to-target distance. Coverage percentage is denoted as c_p further and by its definition, it takes values in the (0, 1] interval.

As it was told before, in the exponential sensing model and the hybrid sensing model, c_p decreases with increasing distance. The question arises: what will be the coverage percentage at the farthest point to the sensor (i.e. $d(p_s, q) = r$) since it is going to decrease exponentially? As presented in Figure 3.11, c_i denotes coverage percentage at the peak (i.e. the farthest point within the sensing range) where sensor-to-target distance is equal to the sensing range of the sensor. In the thesis, c_i is chosen to be 0.3 for both the exponential and hybrid sensing models.

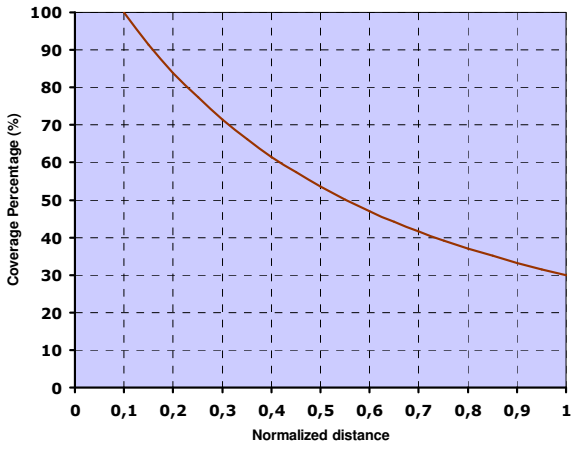


Figure 3.11. Coverage percentage function for the exponential sensing model with $c_i = 0.3$

Additionally, when calculating area coverage for these models, the following approach was adopted. The area of a sub-triangle i is multiplied with its corresponding coverage percentage (c_p). When c_p is equal to 1 (i.e. full coverage), then this multiplication will be equal to the area of the corresponding sub-triangle. As coverage percentage decreases, so will do the result of multiplication. Therefore, we can think of this multiplication as the fraction of area being covered. This will lead us to the following definition.

For the exponential and hybrid sensing models; for a sub-triangle i with an area of $|A_i|$, the amount of covered area can be defined using Equation 3.2.

$$\left|A_{pi}\right| = c_p \times \left|A_i\right| \quad \textbf{(3.2)}$$

We refer to $|A_{pi}|$ as pseudo-covered area. To give an example, let us think of a sub-triangle i with an area of 100 units and let us assume that this sub-triangle is in the range of 2 sensors. One of these sensor nodes (s_1) adopts the binary sensing model, while another one (s_2) adopts the Exponential sensing model (however in the thesis, all sensors are of one type in a scenario). It is obvious that the coverage percentage for s_1 is equal to 1. Furthermore, let us assume that the distance between s_2 and the sub-triangle is such that the coverage percentage for s_2 is equal to 0.5. When we multiply this value with the area of the sub-triangle, we obtain 50. This is equivalent to a case where we have a sub-triangle having an area of 50 units and a sensor with the binary sensing model. It is acceptable to use such an approach because the result of multiplication – pseudo-covered area – will be equal to the average covered area when repeated infinitely many times.

For the binary sensing model, A_{pi} will be equal to the area of the sub-triangle because c_p is equal to 1 by the definition of the model.

Finally, if a sub-triangle is covered by more than one mote, the decision is made so that the amount of covered area of the corresponding sub-triangle is calculated as the largest of these A_{pi} values.

The process is repeated for each sub-triangle and for each of these sub-triangles an A_{pi} value is obtained. In the end, the totally covered area is calculated. In order to calculate f_a, this value is then divided to the total area of the terrain.

3.7.2.2 Line of Sight

A sub-triangle was said to be covered (fully or partially) if it is within the sensing range of a sensor node and if there is a line-of-sight (LoS) between the sensor and a target. Throughout the scope of this thesis, it was assumed that a sub-triangle is covered if and only if there is a straight line starting from a target and ending at a sensor node and it is important that this line is not intersecting with any other object along its path. No signal reflection, deflection or scattering is taken into account. Since thermal and optical sensors act in a similar fashion, this thesis was said to be valid for networks with sensor nodes which are equipped with any of these two sensor types.

A rather simple technique was used to check whether there is a line of sight between two arbitrary points. It is seen from the Figure 3.12 that there exists a line-of-sight between points A and B, and there is no line of sight between points A and C.

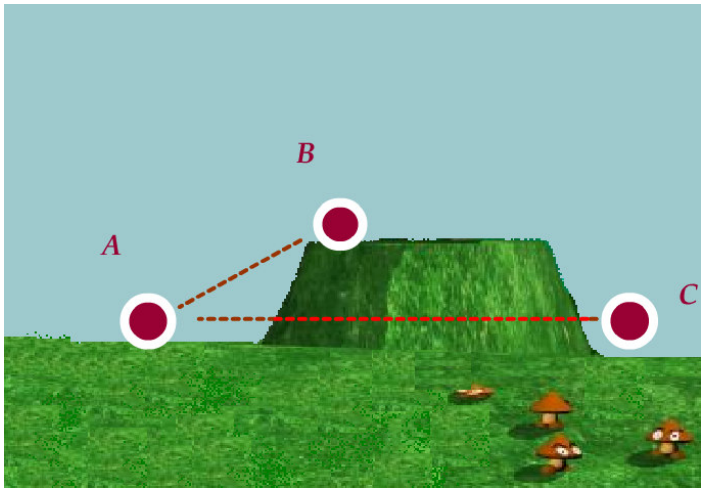


Figure 3.12. Line of Sight Test

Since the TIN model is composed of triangles, a straightforward method would be to calculate the equation of line connecting two points. In the next step a line-triangle intersection test could be carried on for all triangles. In the thesis, terrain models are quite large in size. And after the triangle subdivision step, a terrain was divided into approximately 200.000 sub-triangles. It is obvious that testing a line with all of these sub-triangles is not very practical. To check whether a sub-triangle is in line of sight of a sensor, all of the three endpoints of the sub-triangle need to be checked. For each line connecting the sensor and a point, it is needed to check whether that line is intersecting with any other sub-triangle or not. It is obvious that doing so is not

practical in terms of computation amount. Instead, a rather simple technique is used. Consider a vector connecting points p and q. This vector is divided into n sub-vectors of equal length. Thereafter, by starting from point p and by adding this sub-vector at each step, the height (amount of elevation) at that particular intermediate point is calculated. The calculated value is compared to the elevation amount of the terrain at the same point. If elevation amount of terrain is smaller, then line passes over the terrain. But if terrain elevation is greater than the calculated height value, this means that line passes through the terrain. In the thesis, the sensor height was chosen to be around 10 centimeters and the target height was chosen to be around 60 centimeters.

For a general case, as n goes to infinity, this approach would produce more accurate results. As n decreases, so does the accuracy of this technique. Till this point, all of these are valid for a general case where no matter how many obstacles there are and no matter how long is the distance between two lines. However, in the thesis some assumptions are made and this technique can absolutely exploit these assumptions.

It can be said that a line between a sensor and a point cannot exceed the sensing radius of a sensor which is 25 meters long just because otherwise that point would not be of any interest because it could not be reached. At this step, the question is like the following. Into how many parts is it sufficient to divide this line to be able to carry on the LoS test accurately? Looking over the terrain model may assist in finding the answer to this question.

The TIN model used in the thesis was generated from grid data where the terrain was sampled at every 5 meters and it is certain that the only points where changes can occur are near the sampled grid points. Actually, in this work there is no need to use large n for terrains since no sharp changes occur in relief of the terrain models used. In the thesis, considering that the terrain was sampled at every 5 meters and considering the fact that selected terrain models do not have extreme sharp changes in relief, the value n was chosen so that the LoS check is carried on at approximately every 1 meter. It was also observed that increasing n resulted in serious time consumption.

4. SIMULATION ENVIRONMENT AND RESULTS

4.1 Development Environment

This thesis was developed using the Microsoft Visual Studio 2005 software development environment using the C# programming language. The simulations were run on a PC with "Intel Pentium 4" 3.20 GHz CPU (Central Processing Unit). The system had 1 GB of RAM (Random Access Memory).

4.2 Software Modules

The simulation software consists of 9 different modules. Each module was created to solve a particular part (i.e. sub-problem) of the simulation software. Some of the main modules are explained briefly.

4.2.1 Terrain

This is the most important component of the simulation. It includes the needed data structures to model and methods to implement issues related to coverage calculation.

4.2.2 Geometry

As its name implies, this module was developed in aim to implement geometrical structures and needed calculations. There are three classes defined in this module: Point3D, Triangle3D and Rectangle3D. The point class was used to model points in space and therefore on terrains. The triangle class was used to model triangles generated as a result of Delaunay Triangulation. These triangles form a base to a TIN model. Since each triangle is formed of three points, this class uses three instances of the Point3D class. Finally, the Rectangle3D class was developed to model rectangles in three-dimensional space.

4.2.3 DirectX

The goal of this module is to visually present results to the user. This module was written via the Microsoft's DirectX API (Application Program Interface). In addition

to visual interpretation of the terrain, user interaction was also added to the graphical interface like rotating the camera to get full understanding of the terrain relief.

4.2.4 Main

As its name implies, this is the main module where all other modules are called. Other modules are explained briefly as follows. The module Triangulator implements Delaunay Triangulation algorithm. The module named Constants was designed to store simulation parameters like sensor radius, covered region color and etc. And finally, the Utilities module handles all collateral functions like copying data structures, generating random numbers and etc.

4.3 Scenarios

The simulations were run on three different terrain models, with three different sensing models, three different deployment strategies and five different sensor count parameters which makes a total of 135 unique scenarios. An optimization could be developed in terms of decreasing processing time; however this has not been studied in the scope of this thesis.

Just like other simulations, this simulation also uses random numbers and thus when obtaining results, it is obligatory to run a single simulation scenario for several times and then to take the average of all these values. The question of how many times is it sufficient to run a single scenario depends on the variability of the values. Typically, confidence interval calculations are carried on for this purpose.

It was decided that a run of 10 times would be enough to obtain meaningful values. Since there are 135 unique scenarios, a total of 1350 simulations were run to obtain complete results. Confidence interval calculations were also carried out and for 95% confidence interval it resulted in a change around 1%. Most of the results give the same change so these are not shown on the plots. The parameters which are used as an input to the simulation are given in the following sub-topics.

4.3.1 Terrain Types

As it was told before, terrain models were obtained from the SRTM ftp site [34]. Three terrain models were used to run the simulation. All models represent a real-

world terrain and all of them have a grid sampling interval of 5 meters. The models were visually rendered using the DirectX engine.

The first terrain model is named Smooth since it has fewer obstacles and is smoother when compared to the other two models. The data for this model was taken from the surroundings of Encanto Golf Park, near Phoenix, USA. To put it exactly, this terrain model was extracted from inside of a virtual rectangle formed by two points whose global coordinates are (33° 28′ 44.27″ N, 112° 05′ 25.86″ W) and (33° 28′ 25.08″ N, 112° 05′ 05.01″ W). The screenshot of this terrain model is given in Figure 4.1.

The second terrain model can be easily distinguished from the previous one from the visual point of view. Named Wavy, this model is more uneven when compared to the previous one and as its name suggests, it is a little bit hilly. The model was extracted from inside of a virtual rectangle formed by two points whose global coordinates are (37° 06′ 44.48″ N, 30° 25′ 14.11″ W) and (37° 10′ 41.32″ N, 30° 33′ 14.09″ W). This time, the location was chosen to be Antalya, Türkiye. The terrain looks like as shown in Figure 4.2.

Like the terrain model Smooth, the last model – Rough – was also extracted from a region nearby Encanto Park, Phoenix and is shown in Figure 4.3.

The purpose of going from a flat terrain to a rough terrain is to see how terrain roughness affects coverage results. Adding more obstacles should worsen line of sight and therefore decrease performance and it is interesting to see how coverage changes according to changes in terrain relief.

All of these terrain models were created using a total of 10000 points. These models are square-shaped and there are 201 points on the x-axis and 201 on the y-axis. The distance between consecutive two points is equal to 5 meters which means that all of the terrain models have dimensions of 1 km × 1 km.

In order to be capable of commenting on relief differences of these terrain models it is necessary to calculate some kind of parameter to represent degree of roughness. A simple technique was applied to achieve this goal. For each model, a standard deviation parameter was calculated. In probability and statistics, standard deviation is defined as the measure of the spread from other values.

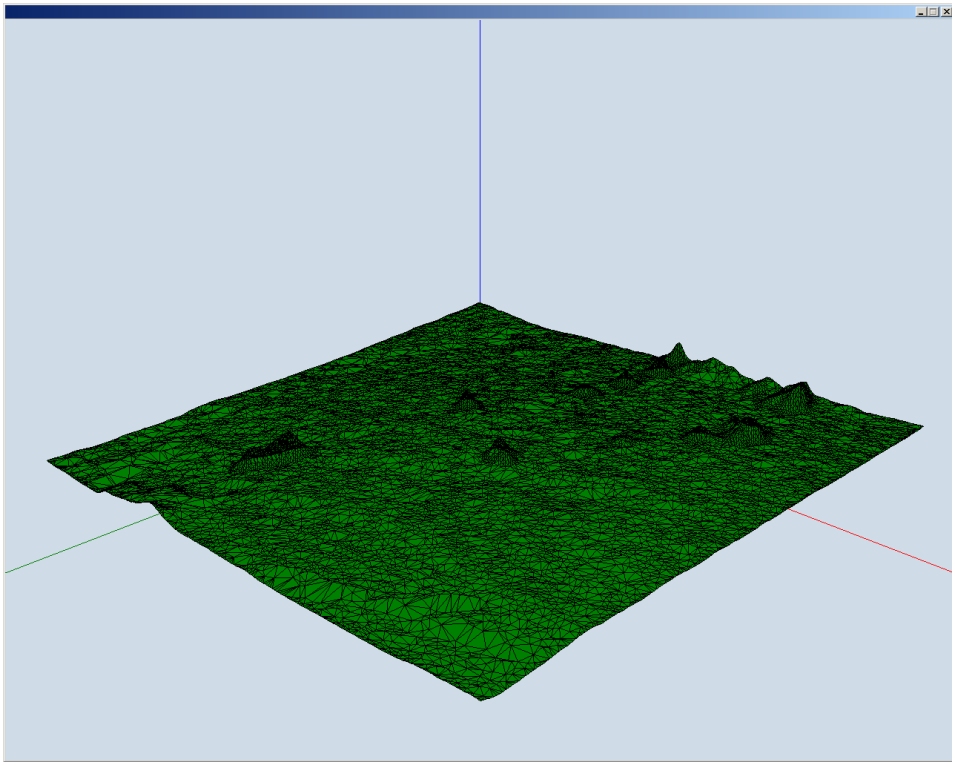


Figure 4.1. Smooth terrain model

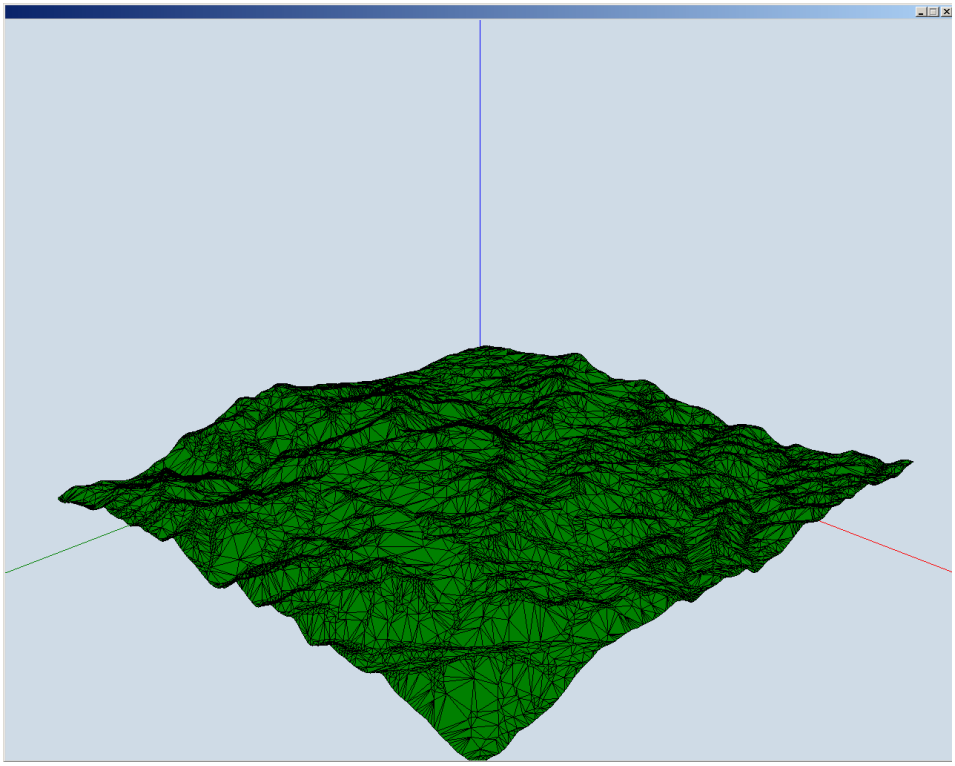


Figure 4.2. Wavy terrain model

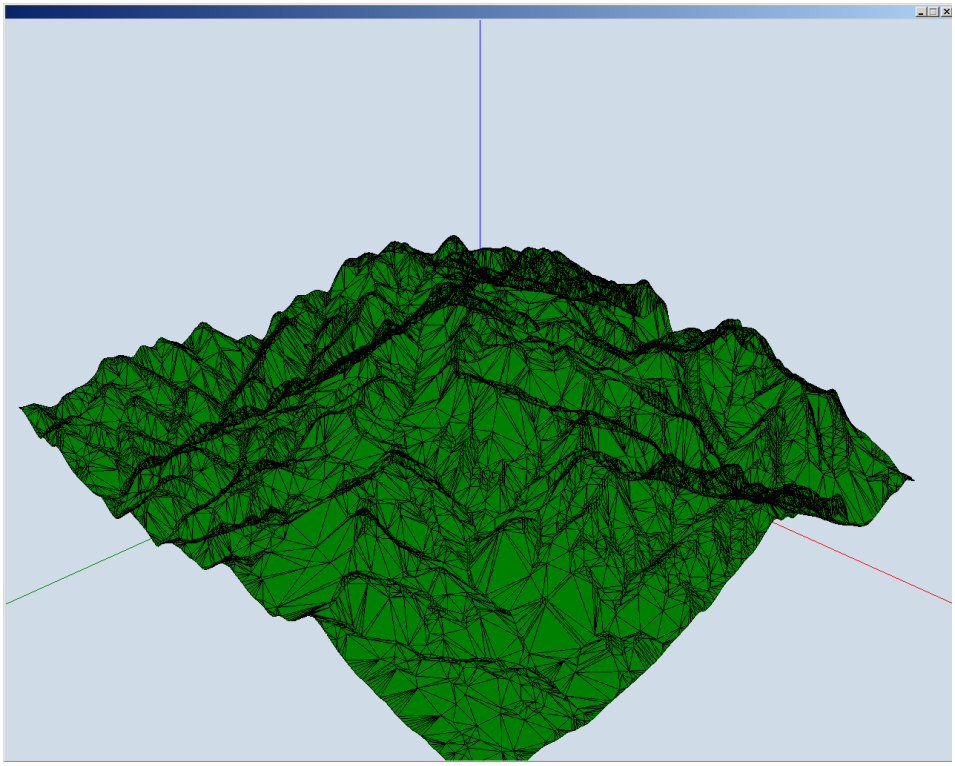


Figure 4.3. Rough terrain model

For each terrain, mean height value was calculated by taking the average of elevation values for all points. Then standard deviation can be calculated using the following formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{4.1}$$

where x_i is the amount of elevation at an arbitrary point i, $\bar{x}$ is the mean of the values and finally, N is the number of points. Standard deviation parameter is a good indicator of the amount of change. In other words, for Smooth terrain model, standard deviation is expected to be less than that of Wavy and Rough. The degree of roughness is shown in Table 4.1 for each of the terrain models.

Table 4.1. The degree of roughness (standard deviation) of each terrain

Terrain	Standard Deviation
Smooth	941,1186
Wavy	1446,5706
Rough	2893,1330

As the terrain roughness increases, so does standard deviation.

4.3.2 Sensor Deployment Strategies

Three different sensor deployment strategies were used in the scope of this thesis.

4.3.2.1 Uniform Deployment

In this deployment strategy, the locations of sensor nodes were determined by generating two uniformly distributed random values – one for vertical coordinate (y-coordinate), another for horizontal (x-coordinate). Using these two values and the information about the terrain model, the elevation amount (i.e. z-coordinate) was calculated at that point. Actually this point is the intersection point of a corresponding triangle and a perpendicular line defined by points (x, y, -∞) and (x, y, +∞). This operation was carried on via the open-source C# library named Sharp3D [46].

A sample uniform deployment with 800 sensors for Smooth terrain model is demonstrated in Figure 4.4.

When looked attentively, it can be seen that nodes are not deployed according to a rule; instead their look seems to be quite random. The implementation of this sensor deployment strategy in the real world is not very easy in terms of feasibility since sensors are thrown from plane in some WSN applications and they may not necessarily conform to the uniform deployment.

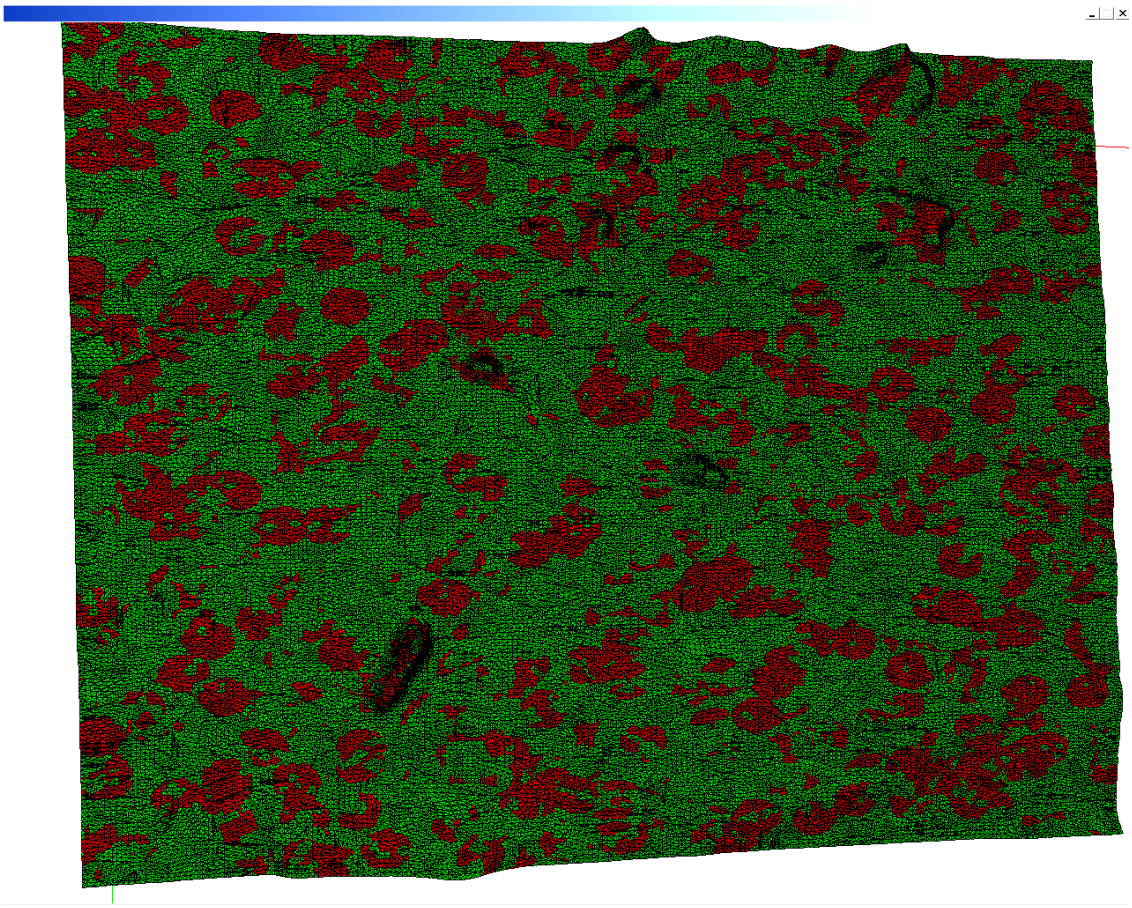


Figure 4.4. Uniform deployment with 800 sensors (Smooth terrain model)

4.3.2.2 Plane Deployment

Another deployment strategy can be arranged as if a plane flew over the terrain and dropped sensors. In Section 4.3.1 it was stated that the dimensions of a sub-square is 5 to 5 meters. Consider a scenario where the terrain is divided into 10 parallel "flight lines" of equal width. In the thesis, this width was taken to be 100 meters. When it comes to deployed sensor node count, total number of sensors is also divided into the number of "flight lines", i.e.10. Then the number of sensors is divided into the length of a "flight line" to determine the interval at which the sensors will be dropped and this interval is decided to be equal. However, at each step on a "flight-line" a normally (Gaussian) distributed random value is generated to decide a deviation amount. This deviation was introduced to make the deployment scheme more realistic as factors like wind can affect final positions of the sensors. If flight lines are vertical, then these deviations are modeled to be horizontal. Moreover, the amount deviation is modeled to be random so that deviation gets smaller as the distance to the plane "flight line" increases as shown in Figure 4.5.

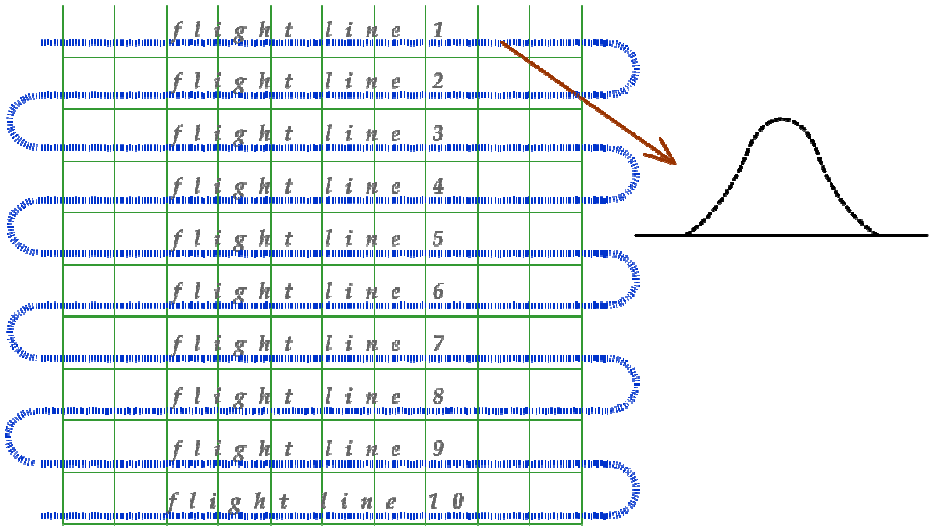


Figure 4.5. "Flight lines" and the amount of deviation modeled with Gaussian distribution with parameters $\mu = 0$ and $\sigma^2 = 1$

A sample deployment with five flight lines for Smooth terrain model is demonstrated in Figure 4.6. For the deployment with 5 flight lines, it can be seen that there exist line-alike shapes. This is quite natural when considering the way the sensors were deployed. However, for a deployment with 10 flight lines, it is not easy to distinguish these lines with unaided eye. A sample deployment with ten flight lines for Smooth terrain model is demonstrated in Figure 4.7.

As the number of flight lines increases, the deployment tends to be more similar to the uniform deployment strategy from the visual point of view.

As for the implementation of this sensor deployment scheme, it can be obviously said that this scheme is more feasible when compared to uniform deployment described in Section 4.3.2.1.

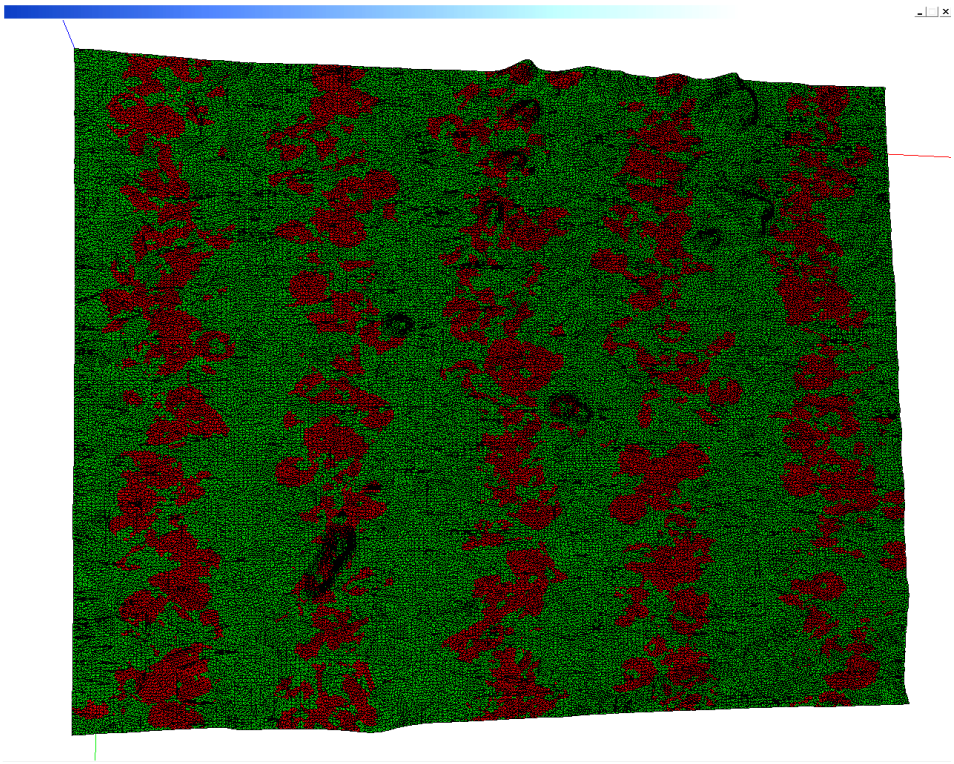


Figure 4.6. The plane deployment with 800 sensors, 5 flight lines (Smooth terrain model)

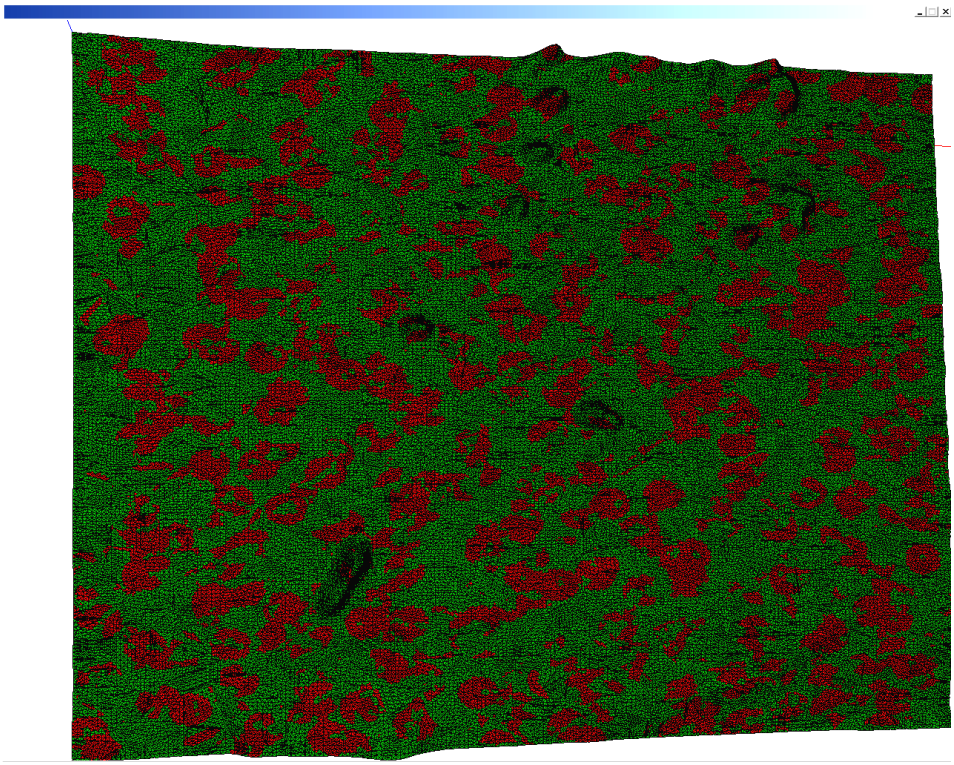


Figure 4.7. The plane deployment with 800 sensors, 10 flight lines (Smooth terrain model)

4.3.2.3 Cross Deployment

The main difference between this deployment strategy and the previous two strategies is that sensor density (number of sensors per unit area) is not the same at all regions within the terrain. In uniform deployment, sensor density is assumed to be equal because as number of sensors increases, the density at each point is assumed to converge to some value. Similarly, in a scenario where sensors are dropped from plane over "flight lines", sensor density can also be assumed to be equal throughout the terrain. However the cross deployment scenario is different than the other two deployment strategies. Sensor density is at its maximum near the center of the terrain and as the distance to the center increases, the sensor density decreases.

The purpose of creating such a deployment strategy can be explained as follows. In wireless sensor networks, it is known that sensor nodes collect sensed data and eventually transmit this data to a sink node which in its turn transmits this cumulative data to a large scale network (e.g. Internet). Since not all sensor nodes are within range of the sink node, the collected data should be relayed over other neighboring nodes to finally reach the sink node. As a result, nodes close to the sink node will consume more energy because they will have to transmit not only their own data, but also sensed data of other neighboring nodes. That is why these nodes should sleep more in order to be able to operate as long as possible. To extend the network lifetime, it would be quite rational to preserve the energy of sensor nodes which are close to the sink node because this region is of critical importance. Consider a scenario where nodes near the sink node have ran out of energy and cannot transmit data anymore. The network becomes not connected and therefore useless. If there is sufficient number of nodes nearby the sink node, (i.e. sensor density is high) then this situation should be exploited in such a way that the network lifetime is maximized. The purpose of the cross deployment is to make the center of the terrain (where the sink node is supposed to be) as dense as possible.

Again, "flight line" (see 4.3.2.2 for more details) concept is used to model this deployment strategy. As it was in the previous deployment scheme, the terrain is divided into "flight lines". However, this time the terrain is divided both vertically and horizontally. The total number of sensors is divided into two. Half of it will be deployed when scanning horizontal "flight lines" and the remaining half will be deployed when scanning vertical "flight lines". The following figure demonstrates a

sample cross deployment scenario where darker regions denote areas higher sensor density. As the tones of green get lighter, the density decreases as shown in Figure 4.8.

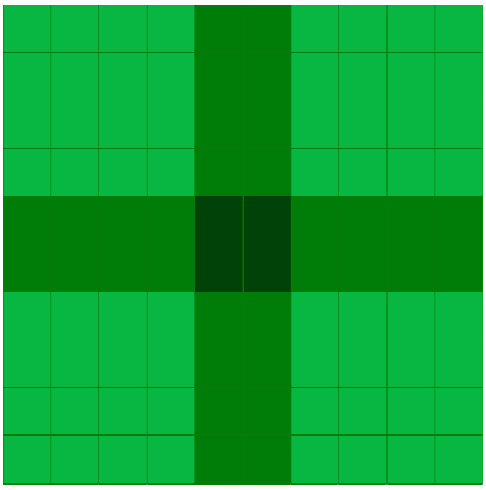


Figure 4.8. Example cross deployment

In the previous deployment strategy, equal number of sensors was deployed for each "flight line". However, in this strategy, they are not equal. Fewer sensors are deployed over the "flight lines" which are far from the center and more sensors are deployed over the lines that are close to the center, i.e. sink node. In the simulations, all flight lines were deployed with the same amount of nodes except two flight lines that are at the center of the terrain (flight lines 5 and 6, as shown in Figure 4.5) as it can be seen from Figure 4.8. If this operation is carried on for both vertical and horizontal directions, then this can yield in diverse sensor densities. This technique can ensure differences in sensor density over the regions within terrain.

The ideal sensor deployment strategy for maximizing network lifetime rather than area coverage would be in the form of rings or circles that start at the sink node and expand their radius with distance. In such a scenario, closest circle to the sink node would have a higher sensor density when compared to the circle that is far away from the sink. It is obvious that in such a scenario, when configured accordingly, the network lifetime parameter can be maximized [47]. However, this deployment strategy introduces its difficulties when it comes to practically implementing it. But the cross deployment implementation is practically feasible.

A sample deployment with 800 sensors for Smooth terrain model is given in Figure 4.9.

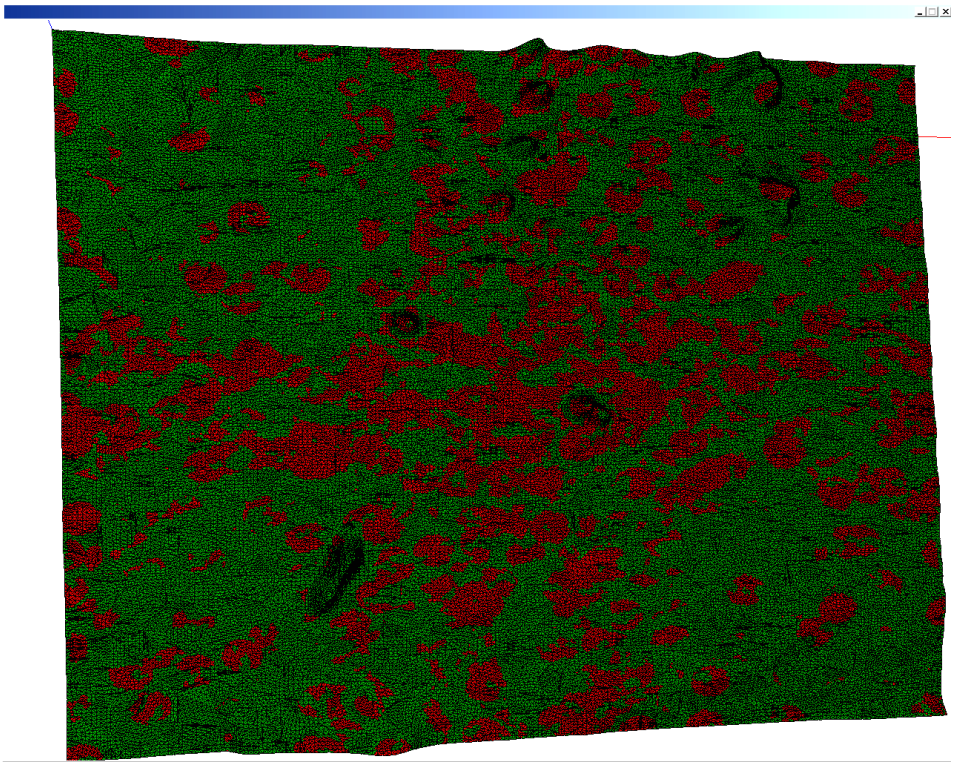


Figure 4.9. The cross deployment with 800 sensors (Smooth terrain model)

When looked attentively, it can be seen that sensor density in the areas close to the center is higher than that of areas that are far to the center of the terrain. Sensors at the center will need to sleep more as they are relay nodes.

4.3.3 Number of Sensors

The simulations were run for five different node count parameters. Starting from 1200 and increasing by 400 the final and maximum node count was decided to be 2800. To put it clearly, the simulations were run for 1200, 1600, 2000, 2400 and 2800 nodes.

4.3.4 Sensing Models

The sensing models used in this thesis are explained in Section 2. To put it briefly, three sensing models were used in simulations: the binary sensing model, the hybrid sensing model and the exponential sensing model. Different sensing models are viewed differently via the graphical interface of the software which is shown in Figure 4.10.

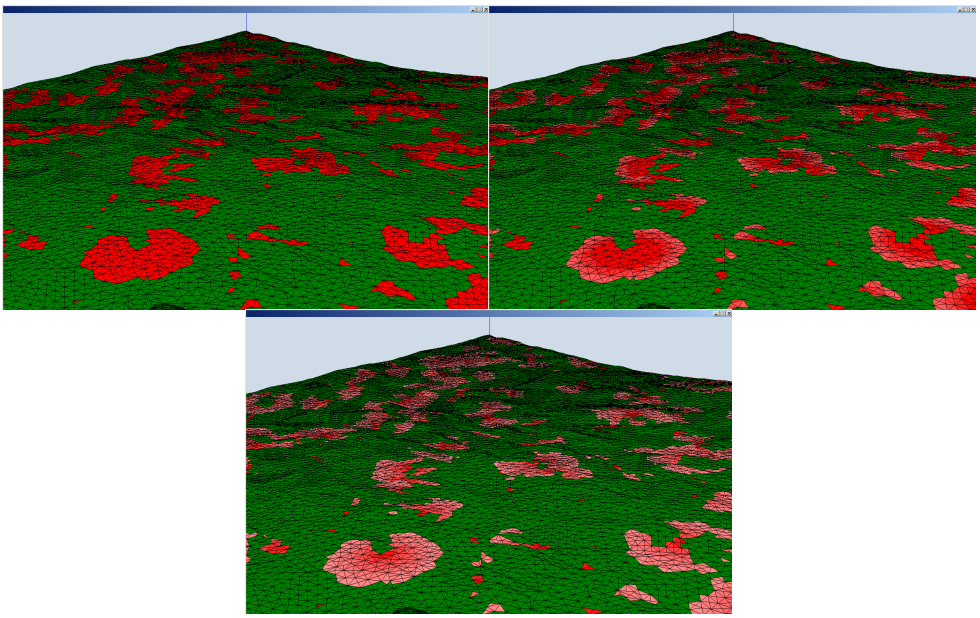


Figure 4.10. Visual interpretation of the binary, hybrid and exponential sensing models

In the binary sensing model, the colors of the sub-triangles are red no matter what the distance to a sensor is. In fact, this approach was selected to demonstrate the coverage percentage. Since in the hybrid and exponential sensing models, the coverage percentage depends on the sensor-to-target distance, the colors get lighter as this distance increases.

4.3.5 Sleeping Scenario

At some point in time, some nodes can sleep without reducing area coverage significantly while the remaining nodes can operate to keep sensing the target. It was explained in Section 4.3.2.3 that in order to make the network live longer, it is reasonable to keep the regions nearby the sink node as dense as possible. It was discussed that to accomplish this task, the cross deployment strategy was introduced. However, it is important to make according calculations and only after that it can be claimed whether one deployment strategy is better than another in terms of conserving energy.

In [47], Subramanian and Fekri obtained upper bounds on the achievable lifetime of the sensor network, independent of routing, sleep scheduling and deployment schemas. This work also analyzes the effects of two sensor deployment schemes on network lifetime. Figure 4.11 and Figure 4.12 demonstrate two scenarios with uniform and non-uniform deployments, respectively.

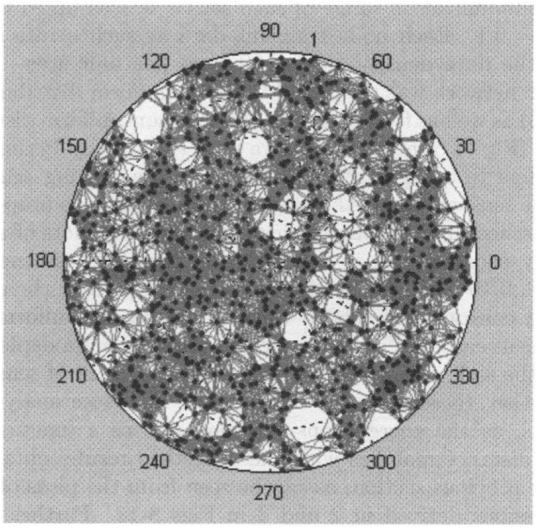


Figure 4.11. Network with uniform deployment of sensors [47]

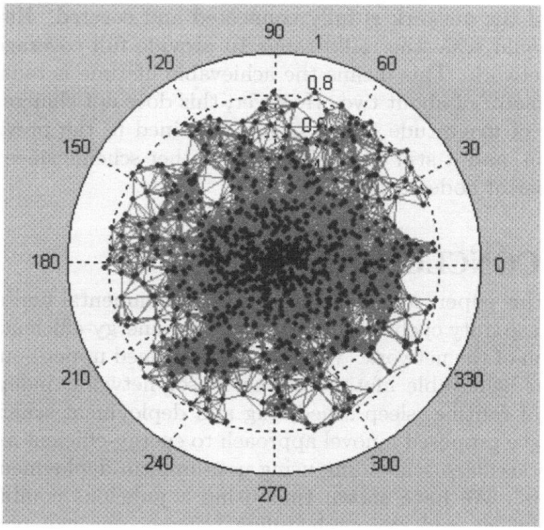


Figure 4.12. Network with non-uniform deployment of sensors [47]

The situation is very similar to the one in this thesis. In the thesis, as a result of uniform deployment, the nodes are located in the similar manner as depicted in Figure 4.11. However, Figure 4.12 reminds the cross deployment where the concentration of sensor nodes is high nearby the sink node (at the center of the terrain). Subramanian and Fekri further analyzed network lifetime for both of these deployment schemes with the use of a sleep-scheduling scheme. The work concludes that with the use of non-uniform deployment as given in Figure 4.12, significant improvements can be obtained in terms of network lifetime.

Furthermore, Guo and McNair analyzed the redundancy versus lifetime tradeoff in [48] and showed that as the sensor density increases, so does the lifetime of the sensor network.

In the context and perspective of these works, a sleeping scenario is described in the following. The terrain is divided into three virtual square-shaped boxes and these boxes are named as the inner box, the middle box and the outer box. In the thesis, the sink node is assumed to be at the center of the terrain. All three of the virtual boxes have a center at this particular point where the sink node is located. When it comes to dimensions of the boxes, the inner box is the smallest one and the outer box is the largest whereas the middle box is in between. After the deployment phase, some of the sensors are located within one of these virtual boxes. A sensor located inside a box, cannot be inside of another one (this implies that there is only one square-shaped box – which is the inner box – and two frame-shaped boxes). All of these boxes are described in Figure 4.13.

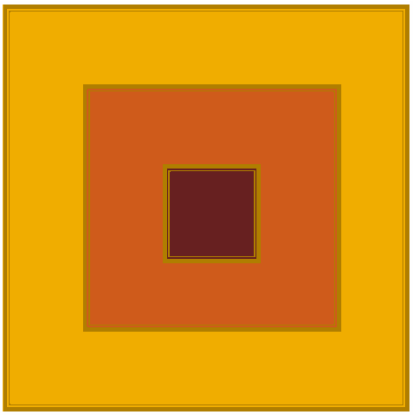


Figure 4.13. Box based sleeping strategy

When compared to the sensor nodes inside other two boxes, the nodes inside the inner box will be closer to the sink node meaning that these nodes will consume more energy. Therefore if sensor density is held high inside the inner box then the nodes inside that box will have more time to sleep, leading to a better chance of extending energy conservation of the network.

The sleeping scenario was implemented using a rather simple approach. Sleeping of a sensor node is modeled in such a way that, at an instant time a sensor is either asleep or awake. To implement this, a sleep probability limit is defined. At an instant time, for each of the sensors a random value between 0 and 1 is uniformly generated

and if this value is smaller than the sleep probability limit, then the sensor node is switched to the sleep mode making it inactive for coverage calculations. Since there are three different regions (boxes), for each of these regions a sleep probability limit is defined. It is clear that the sensors inside the inner box should have a better chance of sleeping when compared to the sensors inside the middle box. Similarly, the same can be claimed for the middle and outer boxes. In this thesis, sleep probability values for inner, middle and outer boxes were chosen to be 60%, 40% and 20%, respectively. The result of a sample sleeping scenario on Smooth terrain model is shown in Figure 4.14.

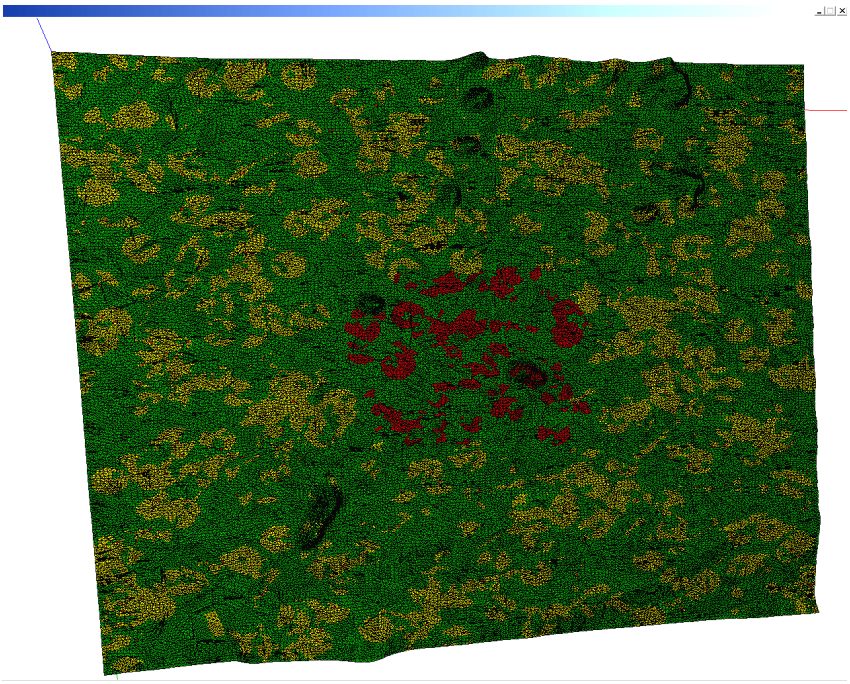


Figure 4.14. The network after the sleep scenario

In Figure 4.14, three boxes can easily be seen. The inner box is red, the medium box is orange and the outer box is yellow. In the perspective of this sleeping scenario, all of the sensor deployment strategies were examined. For each of these three boxes area coverage is calculated before and after sleep and results are compared. Thereafter, some conclusions are given about these results.

4.4 Results

The results of simulations are evaluated over two parameters: area coverage and node coverage. In a sample simulation scenario, firstly the terrain model is created, and then triangulated to form the TIN model. Thereafter, triangles that form the TIN

model are divided into smaller triangles to form sub-triangles. In the next step, sensors are deployed according to the given deployment strategy. Then, area coverage and node coverage are calculated. Then some of the nodes are switched to sleep mode, making them inactive as explained in Section 4.3.5. Area coverage and node coverage parameters are calculated again to measure changes in these parameters. In addition, area coverage inside virtual boxes (see Section 4.3.5) is calculated before and after sleep to see the differences.

Interpretation of the data that is obtained as a result of simulations is carried out in the following form. There are four input parameters for a single scenario of the simulation: sensing model, deployment strategy, terrain type and sensor count. In order to comment on the results, three of them should be kept constant so that the effect of the last parameter can be seen. The x-axis of the figures will represent the number of nodes in the simulation so there are only two parameters left that should be kept constant.

To recall, all data points in the graphs represent the means of ten runs.

4.4.1 Effects on Area Coverage

The effects of simulation input parameters are analyzed to see how these parameters affect the area coverage of the system. Firstly sensing models are considered. Then sensor deployment strategies, terrain types and sensor count parameters are analyzed to see their effects on the area coverage of the system.

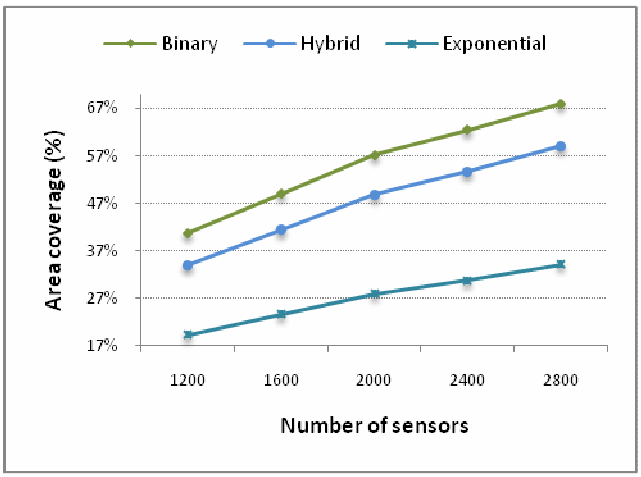
4.4.1.1 Effects of Sensing Models on Area Coverage

Firstly, the effect of sensing models on area coverage will be analyzed. In order to do so, terrain model and deployment strategy parameters are kept constant. The results for uniform sensor deployment scheme and for diverse terrain models are shown in Figure 4.15.

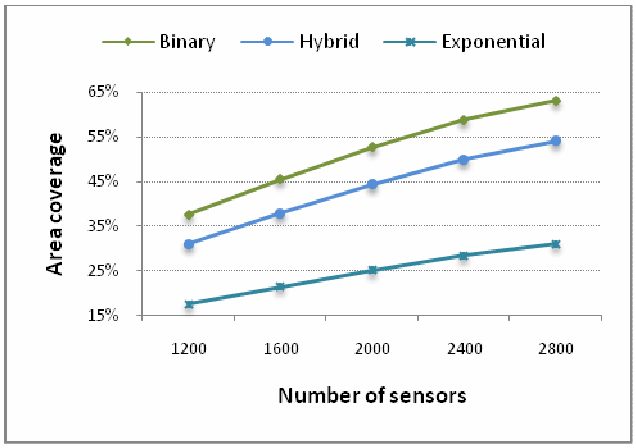
It is seen that the binary sensing model outperforms other two models for uniform sensor deployment scheme. Then comes the hybrid model and in the end comes the exponential sensing model. This is a natural result of these models. The binary model has constant coverage percentage and it is equal to 1 as described in Section 2.1.1. However, other two models have decreasing coverage percentage functions. That is why, the binary sensing model provided the best results.

The graphs for the plane and the cross deployments have nearly the same results; therefore they will not be discussed extensively.

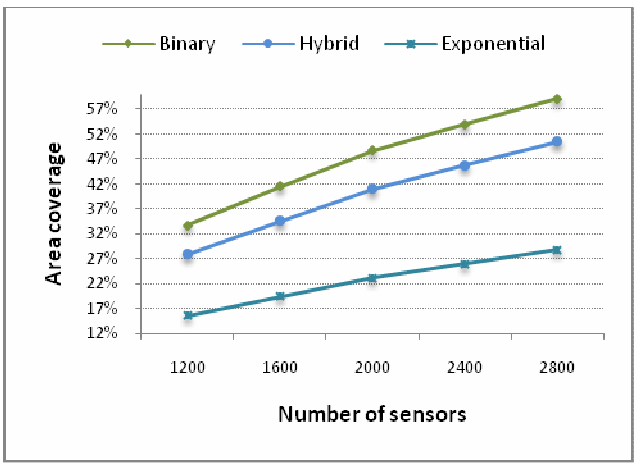
The results for the plane sensor deployment scheme and for diverse terrain models are shown in Figure 4.16.



(a)

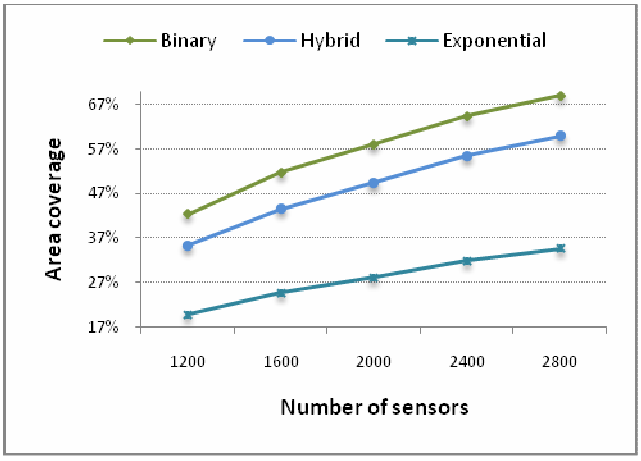


(b)

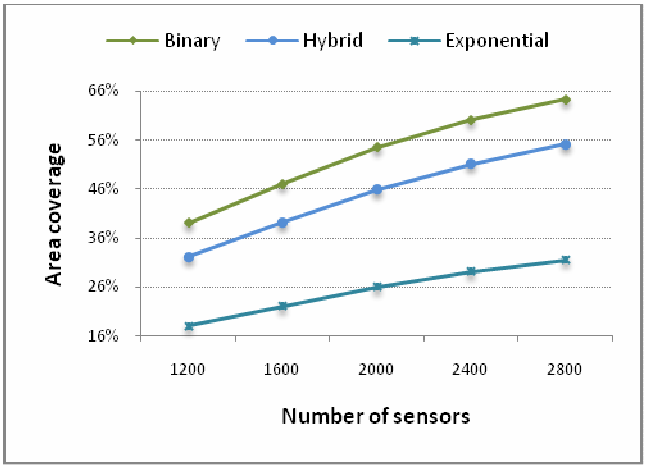


(c)

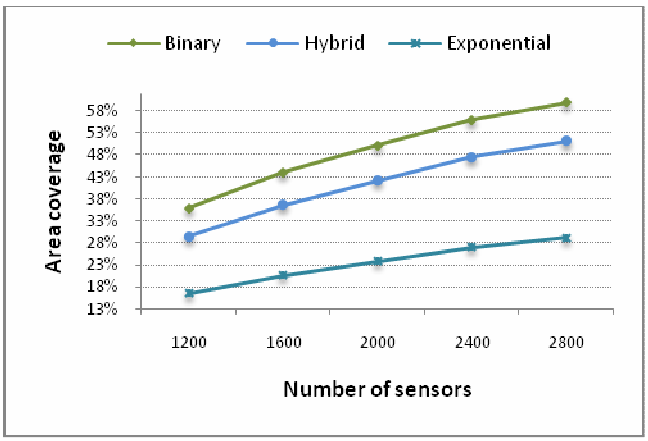
Figure 4.15. Effects of sensing models on area coverage for uniform sensor deployment strategy and (a) Smooth, (b) Wavy and (c) Rough terrain models



(a)



(b)



(c)

Figure 4.16. Effects of sensing models on area coverage for the plane sensor deployment strategy and (a) Smooth, (b) Wavy and (c) Rough terrain models

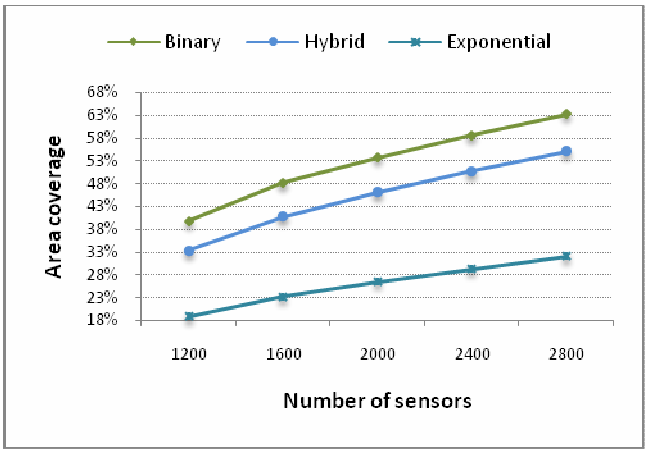
It is seen that the binary sensing model outperforms other two models also for the plane sensor deployment scheme. Then comes the hybrid model and again the exponential sensing model is the last.

The results for the cross sensor deployment scheme and for diverse terrain models are shown in Figure 4.17.

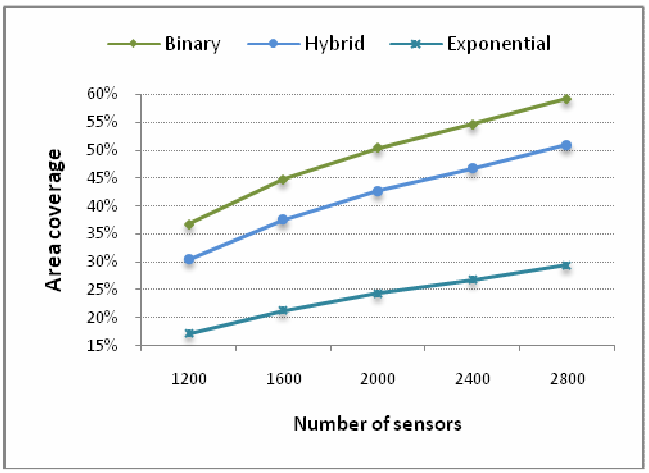
Again, it is seen that the binary sensing model outperforms other two models also for the cross sensor deployment scheme. Then comes the hybrid model and again the exponential sensing model is the last. Again, the reasons were explained above.

In Figures 4.15, 4.16 and 4.17 it is obviously seen that the binary sensing model provides best results in terms of area coverage. However, there is not much difference with the binary model and with the hybrid model, but it is clearly seen that the Exponential model provides worst results when compared to these models. This is due to the fact that the probability of a successfully sensed event decreases exponentially with distance. In the binary sensing model, this probability is independent of distance and for the hybrid sensing model the probability is constant until some point. According to application requirements, one of these sensing models could be selected as the overall network model.

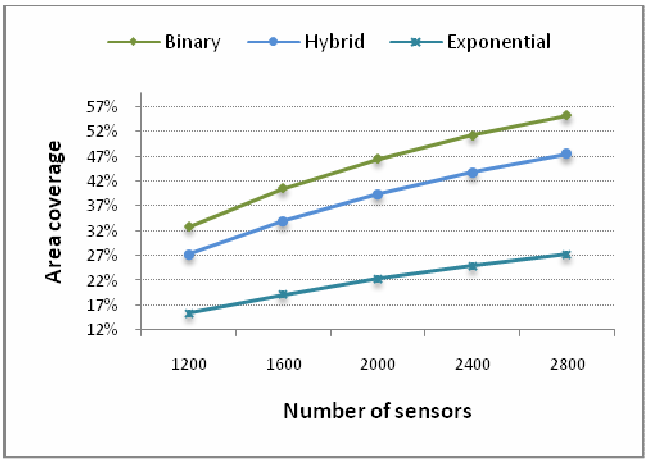
It is seen from Figures 4.15, 4.16 and 4.17 that the relationship between the graphs of three different sensing models is the same. For example, for 2800 sensors deployed in the cross fashion over the Smooth terrain model, the binary sensing model results in approximately 68% of area coverage. Under the same circumstances for Wavy model, the coverage is approximately equal to 64%. Switch from the Smooth model to the Wavy model resulted only in reduced coverage and nothing else was affected by this change. In other words, the area coverage of the hybrid model never exceeded the area coverage of the binary model and similarly the area coverage of the exponential model never exceeded the area coverage of the hybrid model. It can be concluded that, the sensing model directly affects the performance of the network in terms of area coverage and the binary model provides the best results, whereas the exponential model provides the worst results.



(a)



(b)



(c)

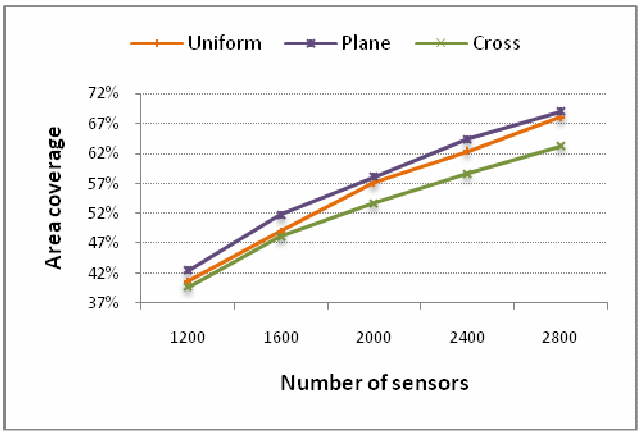
Figure 4.17. Effects of sensing models on area coverage for the cross sensor deployment strategy and (a) Smooth, (b) Wavy and (c) Rough terrain models

4.4.1.2 Effects of Sensor Deployment Strategy on Area Coverage

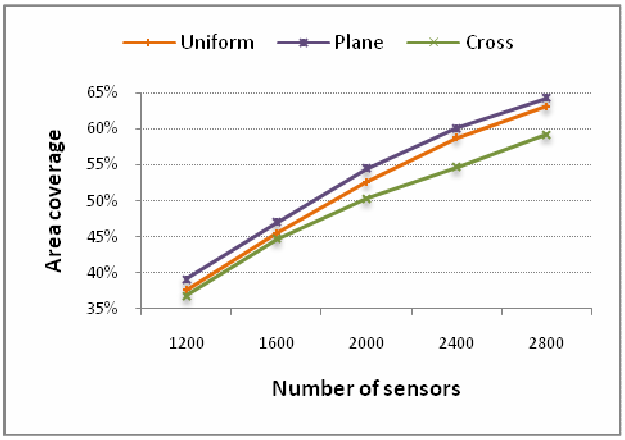
Next, the effect of sensor deployment on area coverage will be analyzed. In order to do so, terrain model and sensing model parameters are kept constant. The results for the binary sensing model and for diverse terrain models are shown in Figure 4.18.

In can be seen from the Figure 4.18 that the plane deployment results in the highest area coverage whereas the cross deployment results in the worst. The purpose of the cross deployment is to make sensor density high at the center of the terrain that will result in high node redundancy at the center but low area coverage at regions far from the center. That is the reason why this deployment provides the worst results in terms of area coverage. It is observed from Figure 4.18 that in all cases studied, the plane deployment is better than the uniform sensor deployment. This is due to the fact that in the plane deployment sensors are scattered deterministically and are deployed so that the deployment is not completely random. The plane deployment is similar to grid deployment. Without deviations (see 4.3.2.2) it would be the grid deployment where sensors are scattered to some specific coordinates (grid points). With the introduction of deviation, the scenario becomes more realistic.

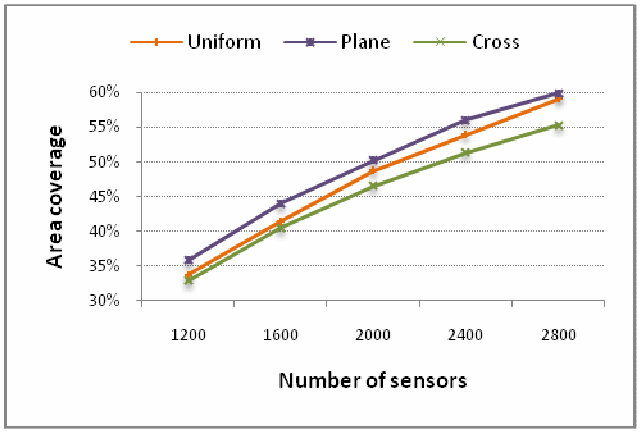
The results for the hybrid sensing model and for diverse terrain models are shown in Figure 4.19.



(a)

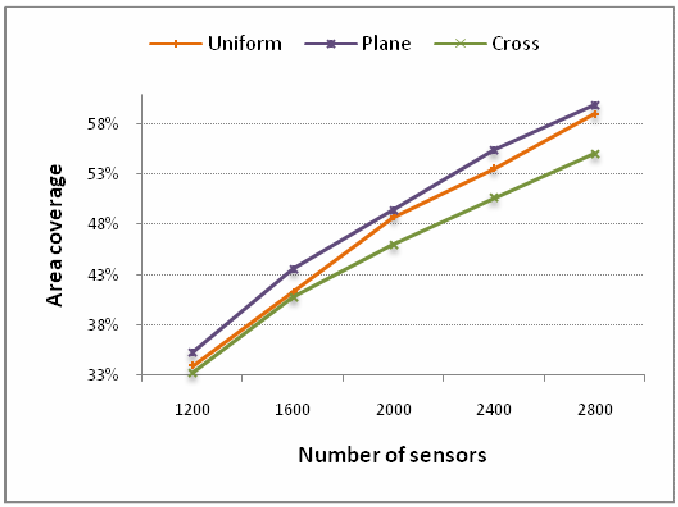


(b)

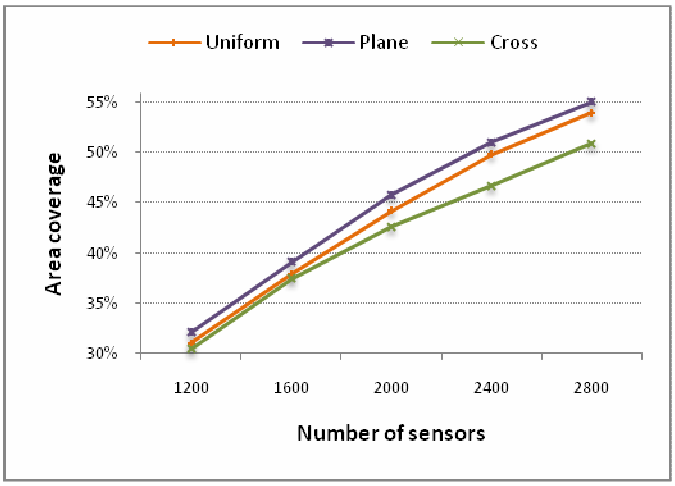


(c)

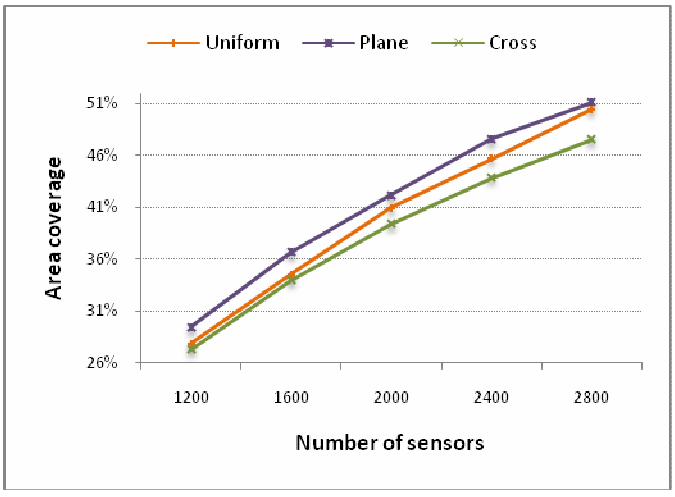
Figure 4.18. Effects of sensor deployment strategy on area coverage for the binary sensing model and (a) Smooth, (b) Wavy and (c) Rough terrain models



(a)



(b)



(c)

Figure 4.19. Effects of sensor deployment strategy on area coverage for the hybrid sensing model and (a) Smooth, (b) Wavy and (c) Rough terrain models

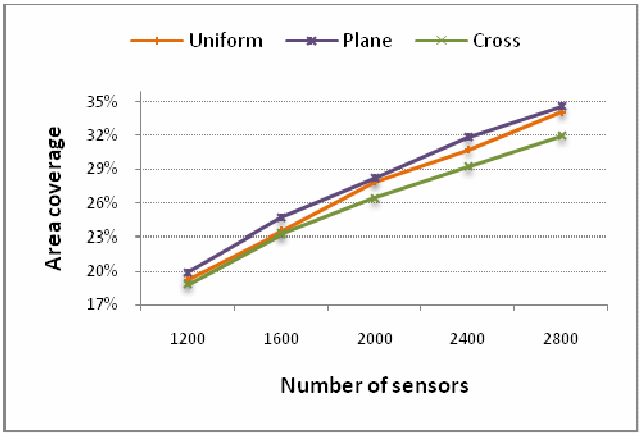
It is seen that the plane deployment scheme outperforms other two schemes also for the hybrid sensing model. Then comes the uniform deployment and again the cross deployment provides the worst results. The reasons that led to these results were explained above.

The results for the exponential sensing model and for diverse terrain models are shown in Figure 4.20.

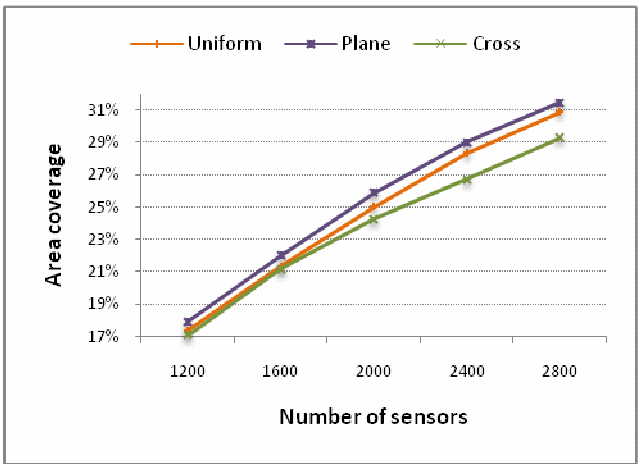
Again, it is seen that the plane deployment scheme outperforms other two schemes also for the exponential sensing model. Then comes the uniform deployment and again the cross deployment is the worst. The reasons that led to these results were explained above.

In Figures 4.18, 4.19 and 4.20 it is seen that the plane deployment provides best results in terms of area coverage because the sensors are deployed so that they do not concentrate over a particular region. On the contrary, the cross deployment provides the worst results since the purpose of this deployment is to make concentration of the sensors high at the center of the terrain. And finally, the uniform deployment provides in-between results.

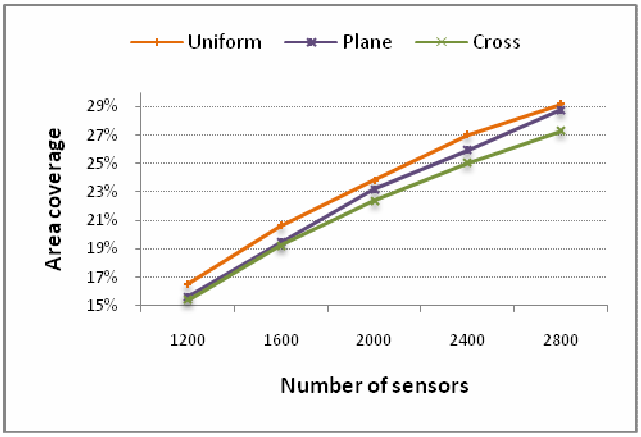
Next, the effect of terrain types on area coverage will be analyzed. In order to do so, sensing model and deployment strategy parameters are kept constant.



(a)



(b)



(c)

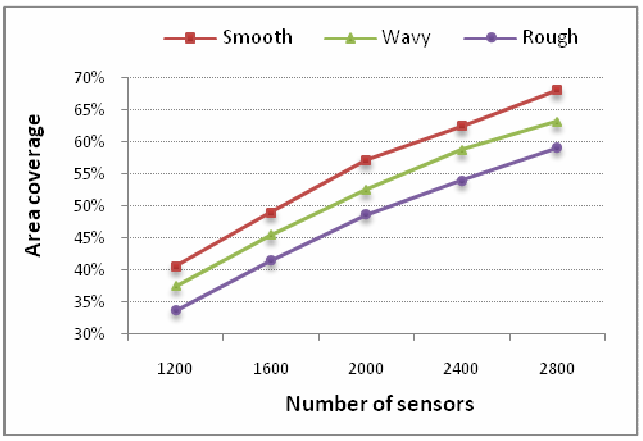
Figure 4.20. Effects of sensor deployment strategy on area coverage for the exponential sensing model and (a) Smooth, (b) Wavy and (c) Rough terrain models

4.4.1.3 Effects of Terrain Type on Area Coverage

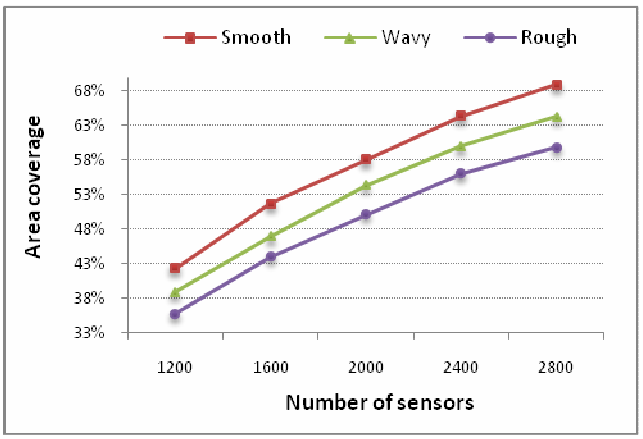
In this section, effects of terrain models on area coverage are analyzed. Firstly, the results for the binary sensing model and for diverse sensor deployment strategies are shown in Figure 4.21.

Since Smooth terrain is the one with the least amount of roughness, it is natural that results for this terrain will outperform other two. As the terrain gets rougher, the LoS suffers and therefore the coverage decreases. Next comes Wavy terrain, and in the end comes Rough terrain.

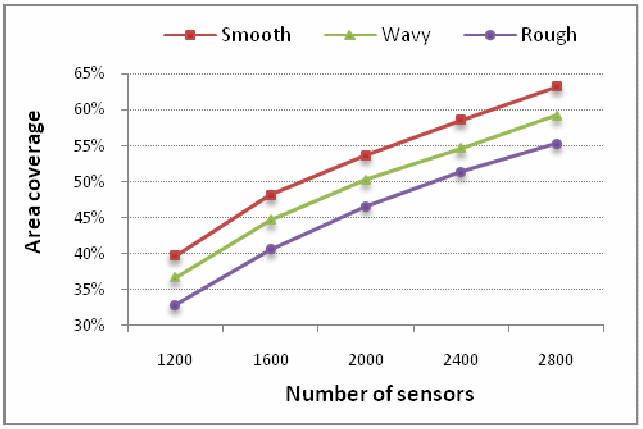
Next, the results for the hybrid sensing model and for diverse sensor deployment strategies are shown in Figure 4.22.



(a)

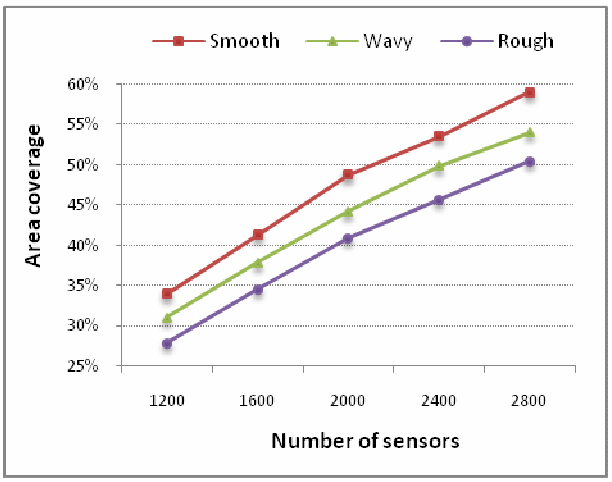


(b)

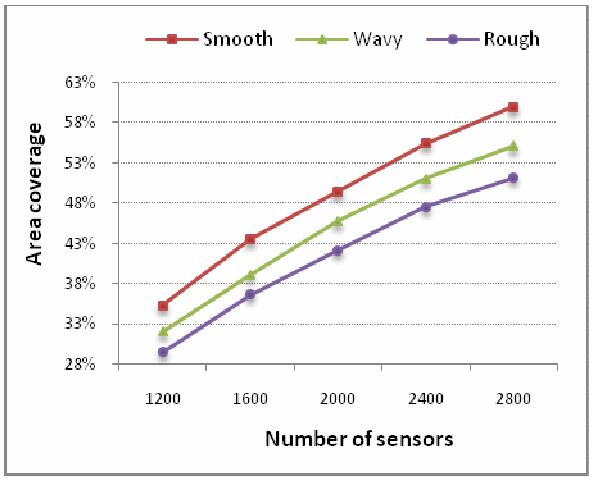


(c)

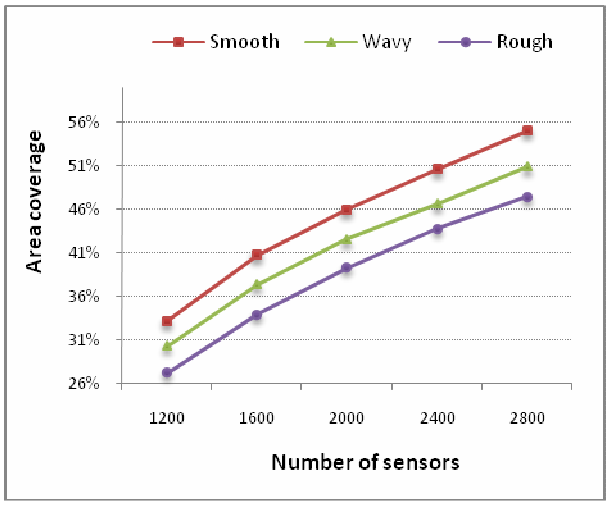
Figure 4.21. Effects of terrain type on area coverage for the binary sensing model and (a) uniform, (b) Plane and (c) Cross sensor deployments



(a)



(b)



(c)

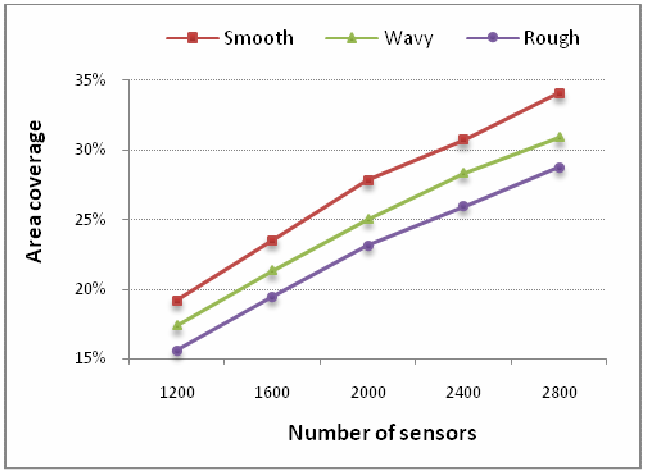
Figure 4.22. Effects of terrain type on area coverage for the hybrid sensing model and (a) uniform, (b) Plane and (c) Cross sensor deployments

Similar results are obtained for the network of sensors with the hybrid sensing model. Since Smooth terrain is the one with the least amount of roughness, again the results for this terrain outperform other two. As the terrain gets rougher, the LoS suffers and therefore the coverage decreases. Next comes Wavy terrain, and in the end comes Rough terrain.

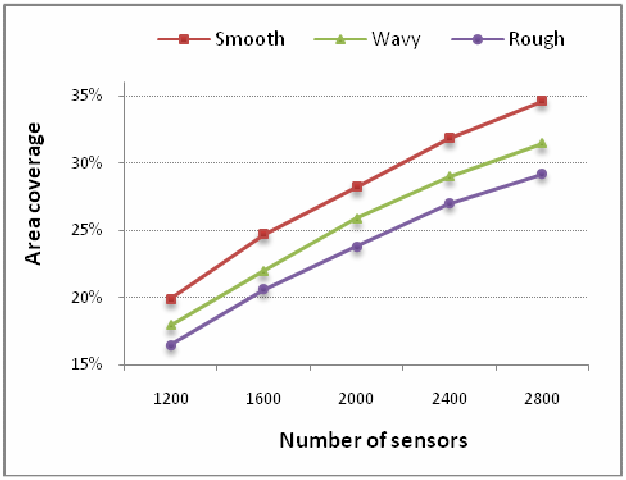
The results for the exponential sensing model and for diverse sensor deployment strategies are shown in Figure 4. 23.

Again, Smooth terrain provides the best results, then Wavy, and in the end comes Rough terrain.

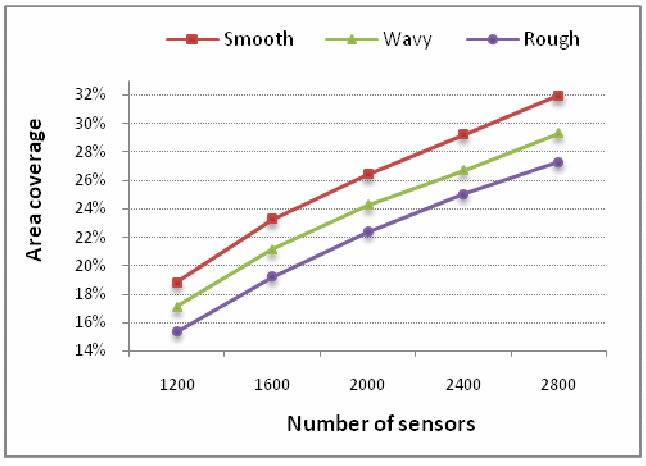
It is clearly seen in Figures 4.21, 4.22 and 4.23 that area coverage gradually decreases as terrain roughness increases. The worst performance is obtained for the Rough terrain model. This can easily be explained because as terrain roughness increases, the line-of-sight suffers and as a result of it coverage suffers.



(a)



(b)



(c)

Figure 4.23. Effects of terrain type on area coverage for the exponential sensing model and (a) uniform, (b) Plane and (c) Cross sensor deployments

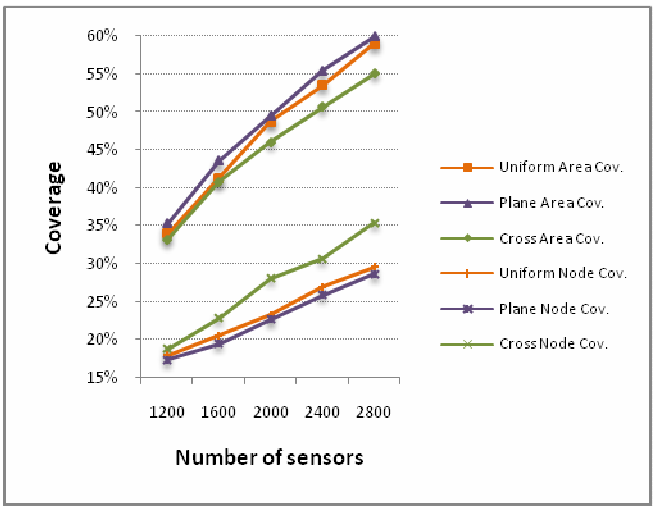
4.4.1.4 Effects of Sensor Count on Area Coverage

From the figures given in Section 4.4.1 it can be observed that as the number of sensors increases, so does the area coverage performance. For each of the unique scenarios, a linear equation can be approximated for making further predictions about the area coverage.

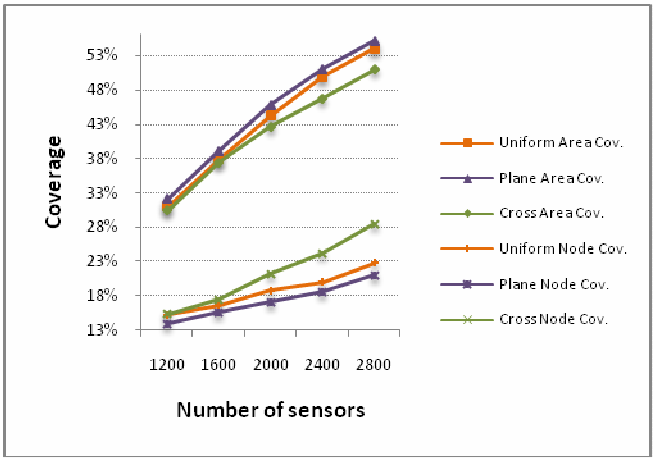
4.4.2 Effects on Node Coverage

Node coverage is a parameter than can only be affected by the sensor deployment strategy. Terrain type also affects it but it is obvious that when terrain roughness increases, line-of-sight will suffer and therefore coverage will decrease as it was described in the previous sections. Since line-of-sight suffers, so will node coverage. Sensing models do not affect the node coverage because there is no such relationship between a sensing model and node redundancy. The only parameter that can affect node coverage is the number of sensors deployed and the deployment strategy. Figure 4.24 demonstrates how node coverage changes according to diverse sensor deployment scenarios.

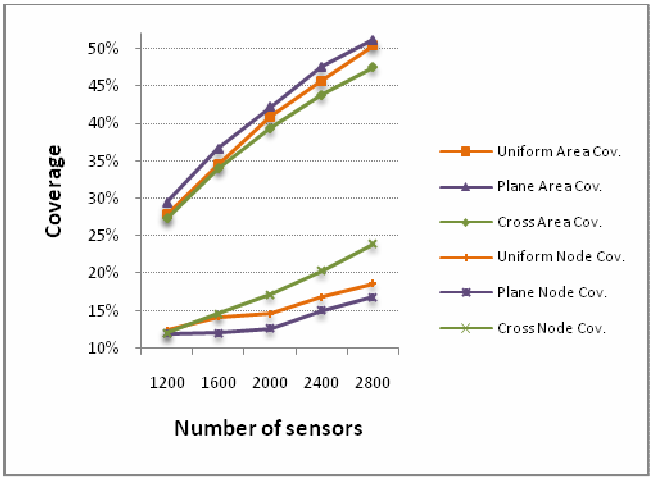
The node coverage parameter is far better in the cross deployment when compared to other deployment strategies because the cross deployment concentrates at the center of the terrain. This results in low area coverage, but high node coverage. As for the plane deployment, the situation is different. In the plane deployment, the nodes are scattered like grid deployment and fewer number of nodes fall near each other thus resulting in low node coverage values. Again, the uniform deployment is in-between of these schemes.



(a)



(b)



(c)

Figure 4.24. Effects of sensor deployment strategies on node coverage for the hybrid sensing model and (a) Smooth, (b) Wavy and (c) Rough terrain models

Naturally, the sensor count parameter also affects node coverage. For instance in Figure 4.24 (c), for the plane deployment the node coverage is almost constant until 1600 sensor count but after this threshold it increases linearly. This can be explained like the following. Consider the first sensor node deployed. Since no other sensors are deployed, the first sensor will contribute to area coverage rather than to node coverage. Then the second node will be deployed and there is a tiny probability that it will fall to a location close to the first node resulting in overlapping coverage. But as the number of sensors increase, after some point the areas that are covered will start to overlap and as a result, node coverage will start to increase. For the case being considered, after 1600 sensors, node coverage increases, which means that a newly deployed sensor is likely to fall and overlap with another previously deployed sensors simply because there are lots of sensors already deployed to the terrain.

4.4.3 Effects on Possible Energy Consumption

The work explained in Section 4.3.5 does not implement sleeping or routing algorithms; it just tries to give insight about how sensor densities may be after switching some nodes to the sleep mode in an instant time. Therefore the results obtained here may not perfectly overlap with the results of a work where sleeping and routing algorithms are implemented. However, the results are expected to be similar.

When considering the factors that may extend the lifetime of a wireless sensor network, it was shown in [48] that sensor node density is of a special importance. In other words, there should be sufficient number of nodes inside a region to make node redundancy high and with the use of appropriately designed sleep algorithms the conserved energy may be maximized [47]. Moreover, node connectivity is another vital factor that must be considered when trying to maximize network lifetime. There is no meaning of achieving quite large coverage if the nodes even cannot communicate with each other simply because they cannot reach each other.

To recall, in this thesis, the terrain is divided into three regions namely inner box, middle box and outer box as described in Section 4.3.5. The sensor nodes inside the inner box have a 60% probability of being asleep at an instant time, while the nodes inside the middle box have a probability of 40% and the nodes inside the outer box have a probability of 20%. Thus, the nodes nearby the sink node will sleep more than

the nodes that are far away. For each of these regions area coverage is calculated before and after the sleep phase. In order to preserve the connectivity of the network, it would be reasonable to keep the sensor density high at the regions nearby the sink node even after the sleep phase. If for instance, the inner box has a very low area coverage value, then most probably, the nodes inside the inner box will not be able to transmit their own sensed data or relay sensed data of other nodes to the sink node no matter how large is the area coverage of the whole terrain. In this context, the area coverage inside the inner box is of a critical importance. The amount of area coverage inside the middle box has a lower importance than that of the inner box, but has a higher importance than that of the outer box.

The results for the binary sensing model, Smooth terrain type and for different number of sensors are shown in Figure 4.25.

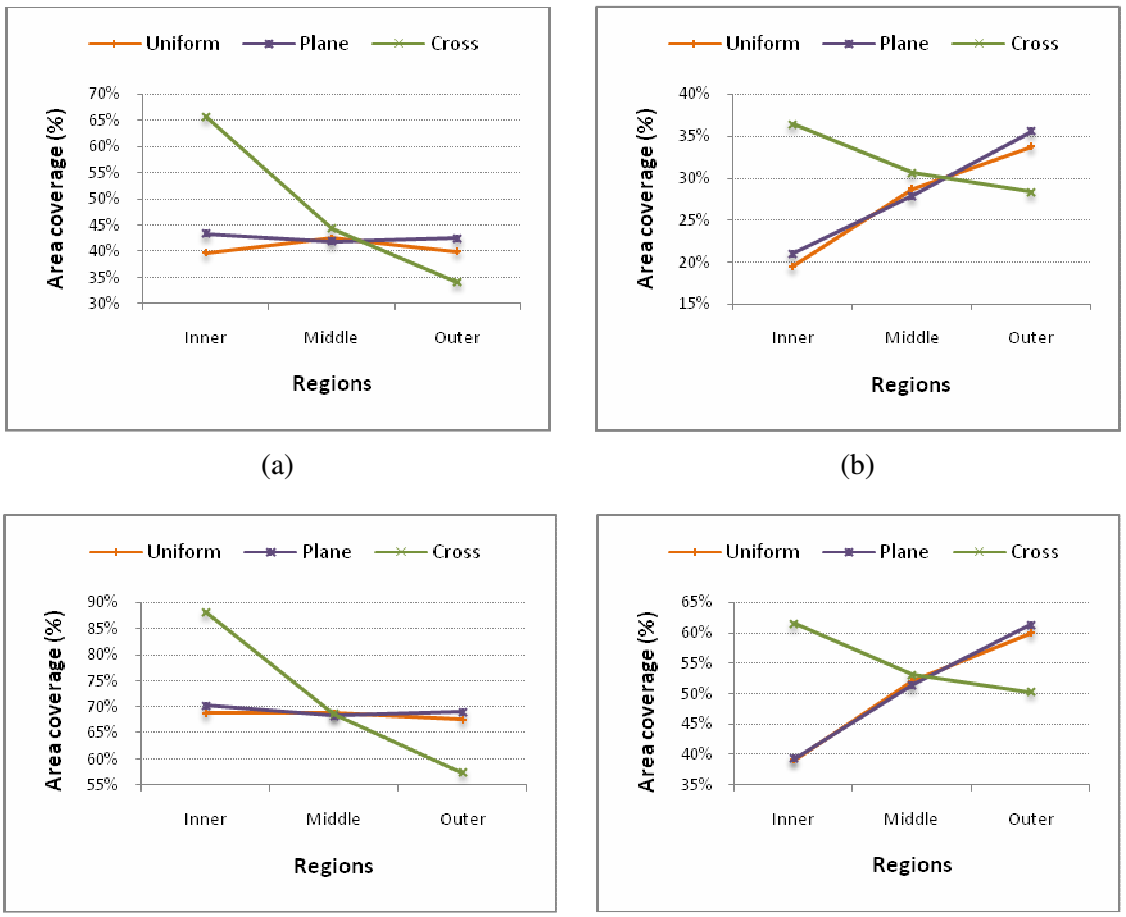
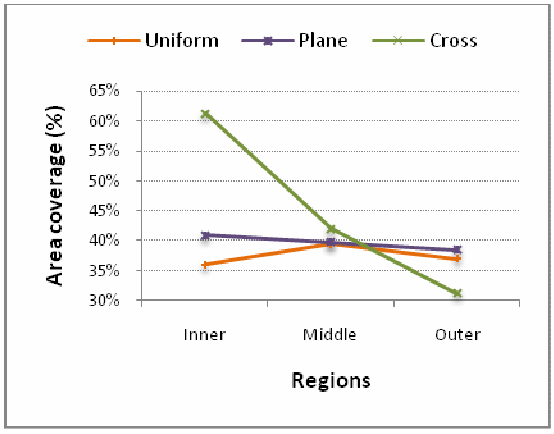


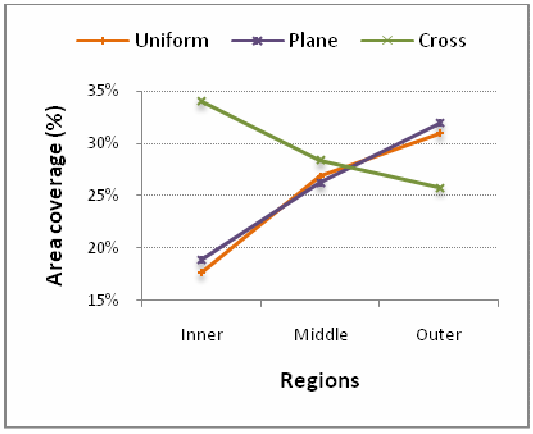
Figure 4.25. Effects of sensor deployment strategies on area coverage of inner, middle and outer boxes for the binary sensing model, Smooth terrain type and (a) 1200 sensors before the sleep period, (b) 1200 sensors after the sleep period, (c) 2800 sensors before the sleep period (d) 2800 sensors after the sleep period

It is seen that even after the sleep phase, area coverage for the cross deployment is higher than that of the plane and uniform. If appropriate sleep scheduling algorithms are used, this can prolong the network lifetime as explained in [47].

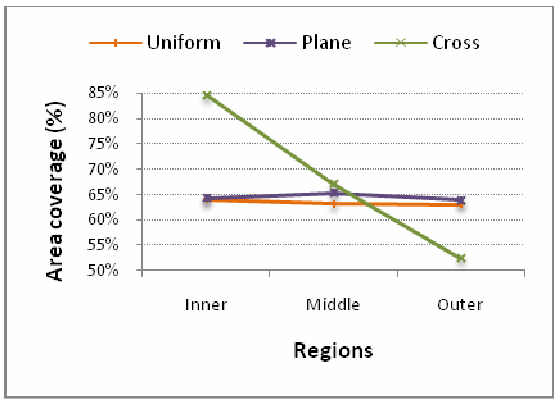
The results for the binary sensing model, Wavy terrain type and different number of sensors are shown in Figure 4.26.



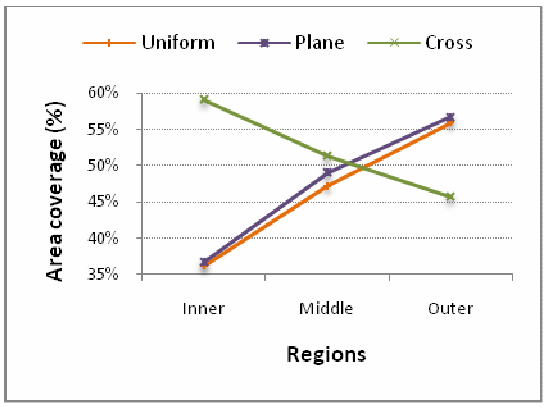
(a)



(b)



(c)



(d)

Figure 4.26. Effects of sensor deployment strategies on area coverage of inner, middle and outer boxes for the binary sensing model, Wavy terrain model and (a) 1200 sensors before the sleep period, (b) 1200 sensors after the sleep period, (c) 2800 sensors before the sleep period (d) 2800 sensors after the sleep period

The comments made about Figure 4.25 for Smooth terrain model can be made about Figure 4.26 which is for Wavy terrain model. It is seen that even after the sleep phase, area coverage for the cross deployment is again higher than that of the plane and uniform deployments. It was said before that if appropriate sleep scheduling algorithms are used, this can prolong network lifetime.

The results for the binary sensing model, Rough terrain type and different number of sensors are shown in Figure 4.27.

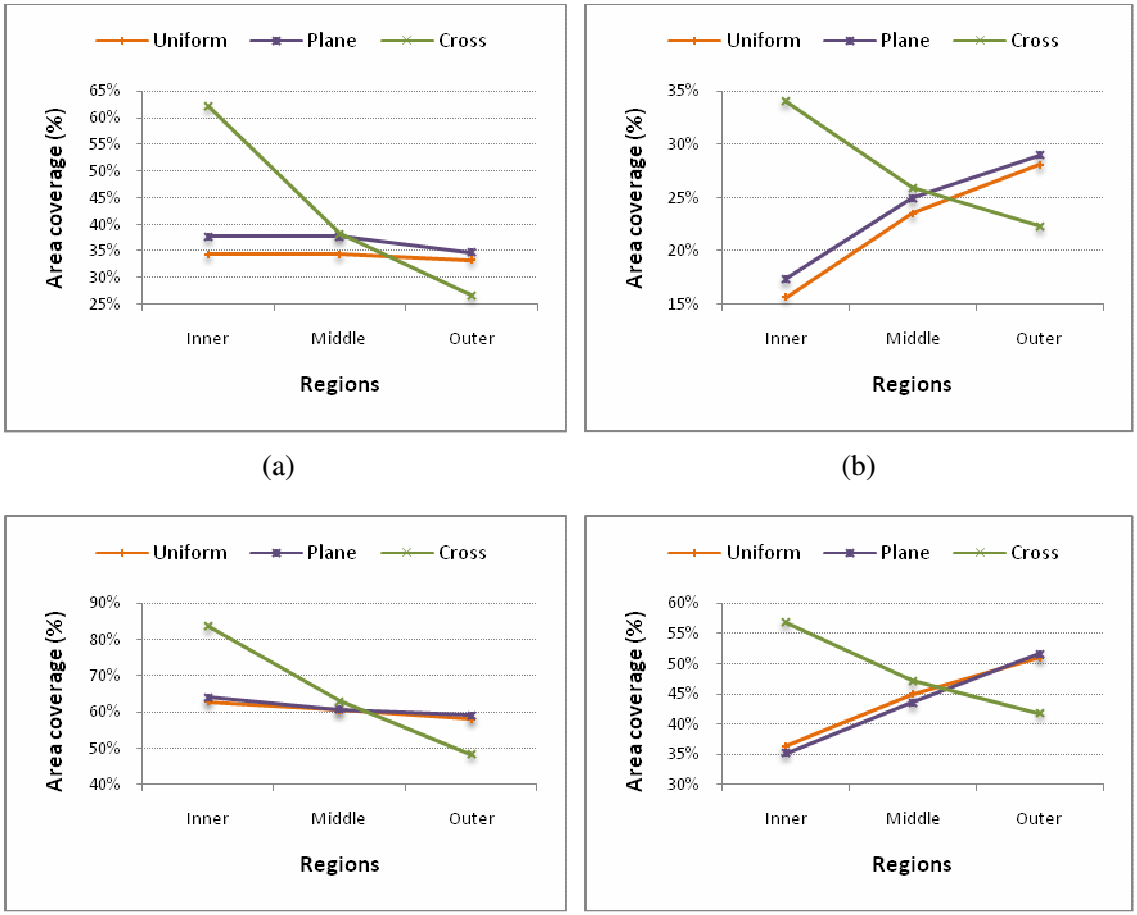


Figure 4.27. Effects of sensor deployment strategies on area coverage of inner, middle and outer boxes for the binary sensing model, Rough terrain model and (a) 1200 sensors before the sleep period, (b) 1200 sensors after the sleep period, (c) 2800 sensors before the sleep period (d) 2800 sensors after the sleep period

The comments made about Figure 4.25 and Figure 4.26 are also valid for Rough terrain model. Therefore, it is expected that using the cross deployment may provide a better chance of prolonging network lifetime but this issue still needs to be justified using sleep scheduling and routing algorithms.

Even if the area coverage of the whole terrain is large, it may not be sufficient for the network to operate correctly because of the surroundings of the sink node may not be dense enough to transmit collected sensing data to the sink node. In other words, the area coverage of the whole terrain is an important factor but the amount of covered area inside the inner, middle and outer boxes is more important than the whole area coverage. When this effect is not considered, the network may run out of energy without being able to transmit the important sensed data.

5. CONCLUSION

This work was carried out to analyze coverage in wireless sensor networks where each sensor is equipped with either thermal or optical sensors. After giving insight about coverage and other performance-related parameters, obtained results were given in Section 4. Furthermore, comments were made on these results and explanations were provided. To sum up, the following conclusions can be made.

From the sensing model point of view, it was observed that the binary sensing model resulted in the best performance in terms of area coverage. The second best was the hybrid model; however it was not very different than the binary model while the results pertaining to the exponential model are the worst. The differences between area coverage parameters were not that much and it was seen that these changes were proportional to each other. Additionally, the area coverage values for different deployment strategies were close in the exponential sensing model so for applications where only this kind of sensor is available, it can be good news.

From the deployment strategy point of view, it was observed that the best area coverage was observed when the plane deployment method was used. However, node coverage suffered significantly under this deployment strategy. Uniform deployment resulted in average results for both area and node coverage. Finally, the cross deployment yielded in the worst area coverage, but in the best node coverage. This is also an important issue since node redundancy can affect the network lifetime directly. Again, according to application requirements, one of these strategies can be selected at the deployment phase. If coverage area of a region is more important than node redundancy, then the plane deployment can be selected. Otherwise, the cross deployment would be an appropriate selection.

From the sensor count point of view, it was observed that this parameter affects both area and node coverage directly. Moreover, it was seen that the increase in node count did not improve area coverage with the same amount, e.g. doubling sensor count did not double area coverage. The same can be said for node coverage.

And finally from terrain type point of view, it was obvious from the beginning that coverage will suffer as terrain roughness increases because it would affect line-of-sight negatively.

5.1 Future Work

This work formed a basis for future works. New modules and new features can be added onto it to implement different aspects of coverage in wireless sensor networks.

To recall, the main assumption of this work is that sensor nodes can only cover points that are in line-of-sight of the sensors and points that do not satisfy this condition were ignored. That is why this work is valid for the network of nodes with thermal and optical sensors. In practice, there exist many other types of sensors and some of them consider signal reflection, deflection, scattering, etc. As a future work, the work implemented in this thesis can be carried out again considering these factors for different types of sensors. Interesting results such as covering the same terrain with less number of sensor nodes can be revealed. Additionally, Fresnel zones that affect LoS can also be considered and carried out in the scope of this thesis to make the system more realistic.

New applications can be implemented by building onto this model. For example, intruder detection algorithms may be analyzed again; this time in three-dimensional space. Sleeping scenarios including duty cycles can be implemented to obtain precise results about energy consumption or network lifetime.

The relationship between area coverage and number of sensors can be approximated as a linear equation. This can provide invaluable information and make it possible to predict area coverage.

Terrain coverage is also studied by using the terrain modeling technique described in [49] for the rough terrain. It is observed that with that model, number of sub-triangles is reduced nearly 10% (when 15 m of triangle perimeter is used with our model). Moreover, coverage is reduced nearly 3%. It can be explained that our model may miss some details due to triangulation done at the beginning. As a future work, we plan to repeat all the scenarios using the model in [49].

The resolution of terrain models can even be adjusted accordingly to achieve better accuracy. Additionally, textures representing physical obstacles such as trees, rocks,

plants can be included as a part of a terrain model which will also make model more realistic.

Briefly, all algorithms that are developed for two-dimensions and all simulations that are carried out in two-dimensional space can be transformed to a new space with one more dimension thus resulting in a better estimation of real-world entities.

REFERENCES

[1] **Akyildiz, I.F., Su, W.L., Sankarasubramaniam, Y., and Cayirci, E.,** 2002. A Survey on Sensor Networks, *IEEE Communications Magazine*, August, pp. 102-114.

[2] **Huang, C.F., Tseng, Y.C. and Lo, L.C.**, 2004. The Coverage Problem in Three-Dimensional Wireless Sensor Networks, *Global Telecommunications Conference GLOBECOM IEEE*, 29 December-3 November, pp. 3182-3186.

[3] **Huang, C.F. and Tseng, Y.C.**, 2003. The Coverage Problem in a Wireless Sensor Network, *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 115-121.

[4] **Watfa, K.M. and Commuri, S.,** 2006. A Coverage Algorithm in 3D Wireless Sensor Networks, *1st International Symposium on Wireless Pervasive Computing*, 16-18 January, pp. 6.

[5] **Dingxing, Z., Ming, X., Yingwen, C. and Shulin, W.,** 2006. Probabilistic Coverage Configuration for Wireless Sensor Networks, *Wireless Communications, Networking and Mobile Computing (WiCOM)*, 22-24 September, pp. 1-4.

[6] **Wan, P.-J. and Yi, C.-W.**, 2006. Coverage by Randomly Deployed Wireless Sensor Networks, *IEEE Transactions on Information Theory*, **52,** 2658-2669.

[7] **Wu, C.-H., Lee, K.-C. and Chung, Y.-C.**, 2006. A Delaunay Triangulation Based Method for Wireless Sensor Network Deployment, *12th*

International Conference on Parallel and Distributed Systems (ICPADS), 12-15 July, pp. 8.

[8] **Rahman, O., Razzaque, A. and Hong, C.S.**, 2007. Probabilistic Sensor Deployment in Wireless Sensor Network: A New Approach, *The 9th International Conference on Advanced Communication Technology*, 12-14 February, pp. 1419-1422.

[9] *Sensor Node*. Dec. 2007. Wikipedia. 28 Aug. 2007 <http://en.wikipedia.org/wiki/Sensor_node>

[10] *Eco Mote: Ultra-compact, Low Power, Expandable Wireless Sensor Node*. Aug. 2007. 12 Oct. 2007 <http://www.ecomote.net/index.html>

[11] *Largest Tiny Network Yet: Large-Scale Demonstration of Self-Organizing Wireless Sensor Networks*. 2001. 12 Oct. 2007 <http://webs.cs.berkeley.edu/800demo/>

[12] *IMEC Reports Two Wireless Platforms*. Apr. 2007. 12 Oct. 2007 <http://www.imec.be/wwwinter/mediacenter/en/ESC2007_ExG.shtml >

[13] **Karl, H., and Willig, A.**, 2005. *Protocols and Architectures for Wireless Sensor Networks*, Wiley, Chichester.

[14] **Liu, B. and Townsley, D.**, 2004. A Study of the Coverage of Large-scale Sensor Networks, *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 25-27 October, pp. 475-483.

[15] **Tian, D., and Georganas, N.D.**, 2002. A Coverage-Preserving Node Scheduling for Large Wireless Sensor Networks, *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, September, pp. 32-41.

[16] **Zhao, F., Guibas, L.**, 2004. *Wireless Sensor Networks – An Information Processing Approach,* Elsevier/Morgan-Kaufman, Amsterdam.

[17] **Kazakos, D. and Papantoni-Kazakos, P.**, 1990. *Detection and Estimation*, Computer Science Press, New York.

[18] **Onur, E., Ersoy, C.**, 2007. Breach Paths and Quality of Deployment in the Lifetime of Surveillance Wireless Sensor Networks, *PhD Thesis,* Boğaziçi University, Institute for Graduate Studies in Science and Engineering, İstanbul.

[19] **Elfes, A.**, 1991. Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception, *Autonomous Mobile Robots: Perception, Mapping and Navigation,* **1**, 60-70.

[20] **Dutta, P., and Arora, A.**, 2003. Sensing civilians, soldiers, and cars, *The Ohio State University Department of Computer and Information Science Technical Report OSU-CISRC-12/03-TR66*.

[21] *Sensors and their Characteristics*. 2006. GENETLAB. 29 Nov. 2007 <http://www.genetlab.com/>

[22] *Radar*. 20 Dec. 2007. Wikipedia. Nov. 2007 <http://en.wikipedia.org/wiki/Radar>

[23] **Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., Vora, A., and Miyashita, M.**, 2004. A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking, *Computer Networks,* **46**.

[24] *LIDAR*. 11 Dec. 2007. Wikipedia. 23 Nov. 2007 <http://en.wikipedia.org/wiki/LIDAR>

[25] **Miller, C., and LaFlamme, R.A.**, 1958. The Digital Terrain Modeling – Theory and Applications, *Photogrammetric Engineering,* **24**, 433-442.

[26] **Petrie, G., and Kennie, T.J.M.**, 1990. *Terrain Modeling in Surveying and Civil Engineering,* McGraw-Hill, New York.

[27] **El-Sheimy, N., Valeo, C., Habib, A.**, 2005. *Digital Terrain Modeling: Acquisition, Manipulation, and Applications*, Artech House, Norwood, MA.

[28] **ESRI, ARC-INFO**, 1992. *Surface Modeling with TIN User's Guide*.

[29] **Rottensteiner, F.**, 2001. Semi-automatic extraction of buildings based on hybrid adjustment using 3D surface models and management of building data in a TIS, *PhD Thesis,* Vienna University of Technology, Faculty of Science and Informatics, Vienna.

[30] *Global Land One-kilometer Base Elevation*. 2007. Dec. 2007 <http://www.ngdc.noaa.gov/mgg/fliers/globedem.html>

[31] **Pike, J.**, *Digital Terrain Elevation Data (DTED).* (2000). Retrieved December, 2007, from: http://www.fas.org/irp/program/core/dted.htm

[32] *Earth Resources Observation and Science*. Aug. 2006. Dec. 2007 <http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html>

[33] *Shuttle Radar Topography Mission.* Feb. 2006. 23 Oct. 2007, <http://www2.jpl.nasa.gov/srtm/>

[34] *SRTM3*. 2005. Sept. 2007 <ftp://e0srp01u.ecs.nasa.gov/srtm/version2/SRTM3/>

[35] *Home of 3DEM Software for Terrain Visualization and Flyby Animation.* Sept. 2007 <http://www.visualizationsoftware.com/3dem.html>

[36] **Fowler, R.J., and Little, J.J.**, 1979. Automatic Extraction of Irregular Network Digital Terrain Models*, ACM,* **4**, 199-200.

[37] **Chen, Z., and Guevara, J.A.**, 1987. Systematic Selection of Very Important Points (VIP) from Digital Terrain Models for Construction of Triangular Irregular Networks, *Proceedings, Auto-Carto 8*, pp. 50-56.

[38] **Poiker, T.K.**, *NCGIA Core Curriculum in GIScience*, (1990). Retrieved October 2007, from http://www.ncgia.ucsb.edu/giscc/units/u056/

[39] *Delaunay Triangulation*. Nov. 2007. Wikipedia. 21 Sept. 2007 <http://en.wikipedia.org/wiki/Delaunay_triangulation>

[40] **Bourke, P.**, 1989. Efficient Triangulation Algorithm Suitable for Terrain Modeling or an Algorithm for Interpolating Irregularly-Spaced Data with Applications in Terrain Modeling, *Pan Pacific Computer Conference*, January.

[41] **Kistner, G.**, *Subdividing a Geodesic Face*. 23 Sept. 2007 <http://phrogz.net/CSS/Geodesics/index.html>

[42] **Thanh, V.**, *Triangle Partitioning*. Sept. 2007 <http://acm.uva.es/p/v107/10734.html>

[43] **Meguerdichian, S., Koushanfar, F., Qu, G., and Potkonjak, M.**, 2001.Exposure in Wireless Ad-Hoc Sensor Networks, *Mobile Computing and Networking,* pp. 139-150.

[44] **Huang, C.F. and Tseng, Y.C.**, 2004. The Coverage Problem in a Wireless Sensor Network, *MONET-Mobile Networks and Applications*.

[45] **Liu, B. and Townsley, D.**, 2003. On the Coverage and Detectability of Large-Scale Wireless Sensor Networks, *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks Conference (WiOpt)*, March.

[46] *Sharp3D.Math Math Library*. 2006. Sharp3D Official Website. Oct. 2007 <http://www.codeplex.com/Wiki/View.aspx?ProjectName=Sharp3D>

[47] **Subramanian, R., and Fekri, F.**, 2006. Sleep Scheduling and Lifetime Maximization in Sensor Networks: Fundamental Limits and Optimal Solutions, *The Fifth International Conference on Information Processing in Sensor Networks (IPSN)*, 19-21 April.

[48] **Guo, Y., and McNair, J.**, 2006. Redundancy versus Lifetime Tradeoff Analysis for Environment Monitoring Using Wireless Sensor Networks, *Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS)*, 24-28 April.

[49] **Arslan, M.G., Ersoy, C.**, 2006. Effect of 3D Topographical Surfaces for the Performance Evaluation of Wireless Sensor Networks, *MS Thesis,* Boğaziçi University, Institute for Graduate Studies in Science and Engineering, İstanbul.

BIOGRAPHY

Anar Khalilov was born in Baku, Azerbaijan in 1985. Raised in Baku, he started his education at Arif Huseynzade High School. After 7 years, he earned a chance to carry on his education at the Ali Bayramly Private High School. Upon completing high school, he started studying Computer Engineering in Istanbul Technical University. After obtaining a B.Sc. degree in 2005, he carried on to study at the same university; this time he attended the M.Sc. Program in Computer Engineering. At the same time he worked at VeriPark in 2006 as a computer engineer in his professional career. After completing his thesis and upon obtaining the M.Sc. degree, it is his wish to become a software architecture designer in his professional work career.