Summer 2006

An Operational Model for Mobile Sensor Cloud Management

Indrajeet Kalyankar
Old Dominion University

Recommended Citation

Kalyankar, Indrajeet. "An Operational Model for Mobile Sensor Cloud Management" (2006). Master of Science (MS), Thesis, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/30wy-ff36
https://digitalcommons.odu.edu/ece_etds/390

AN OPERATIONAL MODEL FOR MOBILE SENSOR CLOUD MANAGEMENT

by

Indrajeet Kalyankar
B. E., Visweswaraiah Technological University, India, September 2003

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
August 2006

Approved by:

Dr. James F. Leathrum, Jr (Advisor)

Dr. Lee A. Belfore, II

Dr. Min Song

ABSTRACT

AN OPERATIONAL MODEL FOR MOBILE SENSOR CLOUD MANAGEMENT

Indrajeet Kalyankar
Old Dominion University, 2006
Advisor: Dr. James F. Leathrum, Jr

Mobile sensors provide a safe, cost effective method for gathering information in hazardous environments. When the hazardous environment is either unexplored, such as the surface of Mars, or unanticipated, such as the result of chemical contamination, it is desirable for a system to gather information with a minimal amount of outside control (localization, decision control, etc.) and prepositioned sensors. If one takes a look at the number of the sensors deployed on a scale, at the lower end is the sole, multipurpose sensor unit. The upper end deals with hordes of inexpensive, expendable sensors. In the middle, a cluster of a reasonable number of sensors has benefits. This thesis speaks of the pros and cons of a small set of sensors and reveals how different its behavior is in comparison with systems at either ends of the scale, in terms of coverage, fault tolerance, and cost effectiveness. This thesis presents the concept of a sensor cloud as a tightly coupled, mobile sensor cluster. The primary characteristics of a sensor cloud are defined in terms of connectivity and coverage. Secondary characteristics like degrees of motion, density and fault tolerance are also laid out. The model of a sensor cloud was derived by extracting commonalities from several of the concepts of sensor collections published in the past. The different kinds of cloud management such as mobility, control and power are discussed. To facilitate development, an operational model for the sensor cloud is developed. The model supports development of control algorithms focused on specific operational needs. Several algorithms are presented within the context of the operational model. Actions such as deployment and navigation are discussed in detail at both the individual sensor level and at the cloud level. Performance parameters such as cloud build up and navigation times are compared for the algorithms and presented.

This is dedicated to my parents,
Ashok and Sushama Kalyankar
and my brother, Avadhoot

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my advisor, Professor James Leathrum Jr., who has the attitude and the substance of a genius; he continually and convincingly conveyed a spirit of adventure in regard to research. Without his guidance and persistent help this thesis would not have been possible. I thank the members of my committee for their time and effort. Finally, I thank all the friends I've met over my two years at Old Dominion especially Tara, Mano, Vasu, Sai, Shital, Jomy and Nanda.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Sensor networks are permeating in every sphere of scientific exploration. This thesis lays out a classification system and a model of a sensor collection. It focusses on the development of a model for small sets of tightly coupled mobile sensors, termed a sensor cloud.

1.1 APPLICATION DOMAIN

Wireless sensor networks will be extensively used in the future to extend our capability to monitor and control the physical environment[1][2], that is otherwise hard to access either due to the environment being unknown or hostile such as in the case of remote harsh fields, disaster areas and toxic regions. A network of sensors improve sensing accuracy over single sensors by providing distributed processing of large amounts of sensing information (e.g.,seismic data, chemical intesity data, images or acoustic data). A multitude of sensors can piece collected data together to offer a multidimensional view of the observed environment. In addition to that, a network of sensors can consist of different kinds of sensors (motion detector, chemical odor detector) each suited to a specific task. When an event is detected, the common abilities of the sensors-like movement, can be focussed to close in on the event for better evaluation. Finally, in the face of the loss of a few sensors, a network of sensors can be designed to work with minimal degradation in performance. A lost sensor can be replaced by another one, or the system can continue operation in its absense. As such, a system of a network of sensors can perform better than that of a handful of specialised expensive sensors.

When the subject comes to studying or observing an environment, deployment is a crucial task. One of the primary objectives in deployment is to obtain optimum coverage of the area. Most of the past papers worked on deployment in environments that are either known or are under control [3]. Traditionally, deployed sensors were considered static. In cases of harsh environments, where manual placement let alone proper regular placement is perilous, scattering by

---

Thesis formatted as per IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and typeset in LaTeX aircraft is one possible solution [4]. Consequently, coverage provided by such tardy and random placement is often insufficient for application requirements. The only solution is to flood the field with a multitude of sensors which increases the cost of the system and makes management difficult. Moreover, disaster areas like a chemical spill at a plant or a burning warehouse necessitate injecting sensors from the entrance. In such cases, it is necessary to make use of mobile sensors. Sensor systems containing a mixture of static and mobile sensors have been proposed in which mobile sensors move from dense regions to sparse areas to improve overall coverage. Guilang et.al [5] [6] have used voronoi diagrams [7] to detect coverage holes and heal them by sending in mobile sensors.

Using sensors to monitor systems involves trade offs between the number of sensors employed and the cost of each of them. At one end, a single, all-capable, mobile sensor can be put to use. Advantages associated with it are simple sensor management and no major communication issues. On the downside, the system is stricken with low data collection rates obviously arising out of the fact that the sole sensor cannot sense a large part of the area at frequent intervals. At the other extreme end, a multitude of inexpensive, small mobile sensors can be used to provide full coverage. The central idea in such a system is to get the work done through the sheer weight of numbers. Issues include sound deployment methods for uniform placement, communication, and navigation techniques. Benefits encompass excellent data collection rates implying quick response to events. However, since the mass of sensors has to be completely connected at all times, it places bounds on the area of coverage given the fixed number of sensors. The only solution other than increasing the inter-sensor distances is to apply more sensors. Somewhere in the middle, a small cluster or cloud of sensors can be used, such that it displays partly the advantages that the systems on either end have to offer. This cluster offers intermediate data refresh rates, can cover any part of the scene by navigating, and provides a reasonable amount of communication complexity.

Wireless sensor networks are being designed for wide ranging applications such as habitat monitoring [8][9][10], health care [11], exploring harsh or manually inaccessible environments, object tracking [12] and military surveillance. Instrumenting natural spaces with numerous networked sensors can enable long-term data collection at scales and resolutions that are difficult, if not impossible, to obtain otherwise. The intimate connection with its immediate physical environment allows each sensor to provide localized measurements and detailed information that is hard to obtain through traditional instrumentation methods. Embedded smart sensors have been devised to operate within the human body to compensate for various diseases. Unobtrusive sensors can be designed in studies involving monitoring of plants and animals in field conditions if there is a concern about the potential impacts of human presence. Object tracking involving surveillance is used for protection of personnel in peacekeeping missions. Military applications include shallow water mine sweeping [13], carrier deck FOD disposal, maintenance inspection [14], and ship hull cleaning [15].

1.2 PROBLEM STATEMENT

The problem that this thesis attempts to solve is to identify the underlying structure upon which all the presented ideas stand and seem relevant. Several papers have been published detailing the different ways in which a bunch of mobile sensors can be used. However there is an absence of formalization of existing methods. This thesis tries to fill that void. It tries to develop an accurate and useful semantics (i.e., meanings must be associated with well-formed actions). Efforts have been put in to identify the core problem domains and weed out the irrelevant issues from the relevant ones. The benefits of formalization can be summarized as follows:

- Leads to a better understanding of the developed systems
- Co-relating seemingly unrelated ideas making it easier to compare, contrast and share
- Substitute similar components between systems developed on different lines

The result is an operational model for a subset of sensor clusters characterised by a small set of tightly coupled, mobile sensors.

1.3 SENSOR CLOUD

One of the earliest papers that used the term "clusters" for sensor collections was in a paper by Heinzelman et. al. [16]. A sensor cloud is defined in this thesis as a small set of mobile sensors that operate as a unit. The individual sensors independently discharge their sensor duties. However, they are required to correspond with their near neighbors while in motion to prevent the cloud from breaking apart. The sensor cloud model comprises the sensor cloud in addition to characteristics and actions associated with it. Connectivity among constituent sensors and area of coverage provided are examples of characteristics. Actions related to a cloud refer to the process of building it and moving it across in an environment. High level actions at the cloud level determine actions at the lower sensor level. The actual development of usefully functional systems consisting of large numbers of simple robotic sensors provides ample challenges from both the technical and management perspectives.

1.4 THESIS' CONTRIBUTION

This thesis presents an operational model for sensor clouds. The model definition makes it necessary that the group of sensors stick to restrictions such as not breaking apart and connectivity with neighbors at all times. It details the corresponding features, classifications and characteristics of an arrangement of sensors. It draws out a framework and demonstrates that past work on collections of sensors subscribe to the model. In addition, a specific cloud arrangement is deveoped and alogrithms related to its deployment and navigation are laid out. It is further shown to exhibit the characteristics required of a sensor collection.

As a step further, this thesis also provides a framework for the design of future systems. Using the model explained in the subsequent pages, a user should be able to identify the different problem domains that make up the system that go into achieving the required objective. This work is important because it lays out the groundwork from which others can extend their ideas for future work.

The formalization technique adopted consisted of extracting high-level abstractions from past papers published. By comparing the abstractions, common actions were identified. Further, by restricting the definition of the model to higher levels and giving free reign to low level design, it was made to encompass several concepts of mobile sensors published in the past.

1.5 THESIS STRUCTURE

The rest of this thesis is organised as follows. Chapter 2 discusses related work. Chapter 3 describes the sensor cloud model. Chapter 4 presents a hexagon sensor model within the context of the sensor cloud model. It details two techniques of deployment and translation. Analysis is carried out on the techniques and results are presented. The thesis is concluded in Chapter 5.

CHAPTER 2

RELATED WORK

This chapter first introduces example applications of mobile sensors. It then discusses a classification of mobile sensors. Finally it presents several models of deployment of mobile sensors.

2.1 APPLICATIONS

Numerous papers have been published detailing implementations of specific applications. A few of them are presented here.

Habitat Monitoring

Application Driver for Wireless Communications Technology by Alberto Cerpa et.al. [8]: This paper describes habitat monitoring as their motivating application to deploy densely distributed wireless networks of sensors and actuators. They subscribe to the thought that new system architectures and new network algorithms must be developed to transform the vast quantity of raw sensor data into a manageable stream of high-level data. To address this, they propose a tiered system architecture in which data collected at numerous, inexpensive sensor nodes is filtered by local processing on its way through to larger, more capable and more expensive nodes. Motes shown in Figure 1 are the smallest components of the tiered sensor architecture.

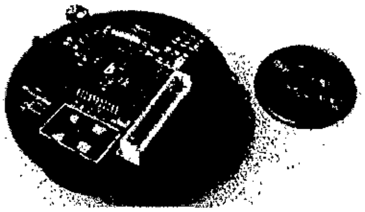


Fig. 1: Mote developed by Kris Pister at UC Berkeley. [8]

Wireless Sensor Networks for Habitat Monitoring by Alan Mainwaring et.al. [9]: This paper provides an in-depth study of applying wireless sensor networks to real-world habitat monitoring. A set of system design requirements are developed that cover the hardware design of the nodes, the design of the sensor network, and the capabilities for remote data access and management. A system architecture

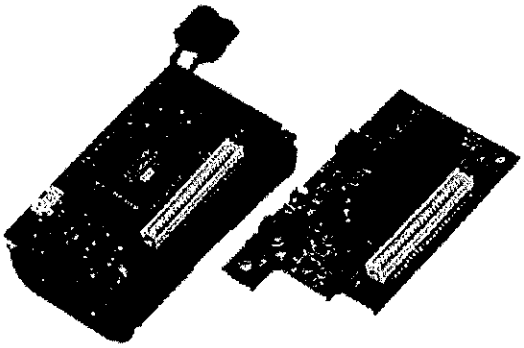


Fig. 2: Mica Unit. [9]

is proposed to address these requirements for habitat monitoring in general, and an instance of the architecture for monitoring seabird nesting environment and behavior is presented. The currently deployed network consists of 32 nodes on a small island off the coast of Maine streaming useful live data onto the web. They use the latest member of the mote family, called Mica, shown in Figure 2.

Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and ZebraNet by Ting Li et.al. [10]: ZebraNet is a mobile, wireless sensor network in which nodes move throughout an environment working to gather and process information about their surroundings. This paper discusses and evaluates ZebraNets system design. The hardware unit they make use of is called a Zebranet unit and is shown in Figure 3.

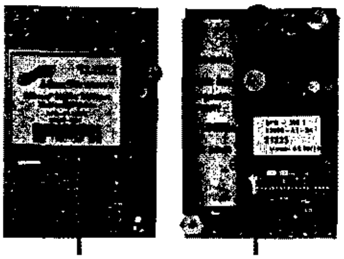


Fig. 3: Zebranet Unit. [10]

Health Care

Research Challenges in Wireless Networks of Biomedical Sensors by Loren Schwiebert at.al. [11]: This paper describes the potential of biomedical smart sensors. It explains the challenges for wireless networking of human-embedded smart sensor arrays and their preliminary approach for wireless networking of a retina prosthesis.

Object Tracking

Object tracking in a multi sensor network by Sebastiaan de Vlaam [12]: This report presents a design and implementation of an object tracking system using a wireless sensor network. This system is able to detect and track objects, and report information about these objects to a central base station. It makes use of Mica2, shown in Figure 4 and developed at UC Berkeley.

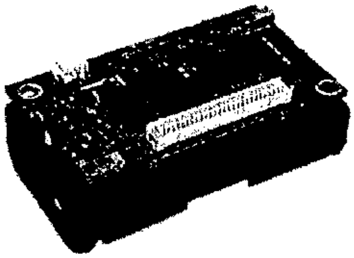
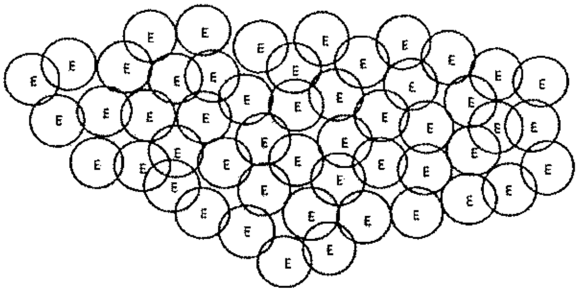


Fig. 4: Mica2 Unit. [12]

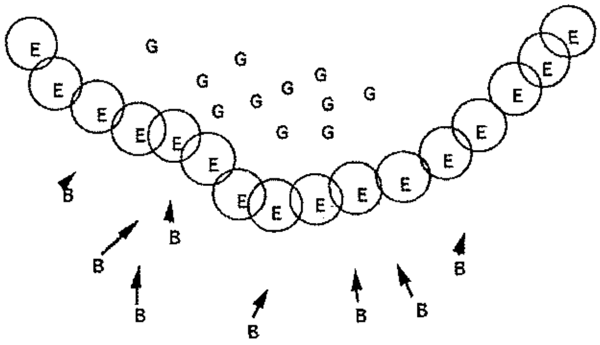
2.2 GAGE'S MODEL

Some papers describe models of mobile sensors performing tasks. The paper of many-robot systems by Douglas Gage [14] [17] describes the concept of mobile sensor nodes moving in groups, performing tasks such as deployment, recovery and providing different type of coverages. These actions can address a broad spectrum of generic applications such as mine deployment, mine sweeping, surveillance, sentry duty, maintenance inspection, ship hull cleaning, and communications relaying. Three varieties of coverage behaviors, graphically depicted in Figure 5, can be distinguished.

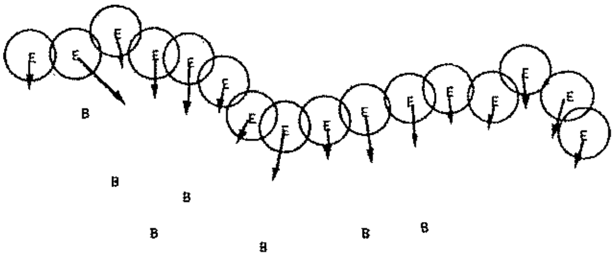
- Blanket coverage: The objective is to achieve a static arrangement of elements that maximizes the detection rate of targets appearing within the coverage area.
- Barrier coverage: the objective is to achieve a static arrangement of elements that minimizes the probability of undetected enemy penetration through the barrier.



a. Blanket



b. Barrier



c. Sweep

Fig. 5: Coverage Behaviors. E, G, and B represent system Elements, ‘Good guys’ to be protected, and ‘Bad guys’ to be engaged, respectively. The circles around system elements represent the effective sensor/effector engagement radius.

- Sweep coverage: the objective is to move a group of elements across a coverage area in a manner which addresses a specified balance between maximizing the number of detections per time and minimizing the number of missed detections per area. (A sweep is roughly equivalent to a moving barrier.)

These behaviors can be used to execute real world tasks. Further, the paper realizes that the key to achieving mission objectives is to ensure that the aggregation of the mobility behaviors exhibited by the individual elements of the system results in the desired behavior of the group as a whole.

2.3 DEPLOYMENT MODELS

Deployment is a critical problem in using sensors extensively. For civilian applications, the sensor network can be installed by hand on the ground or dropped from an unmanned aerial vehicle (UAV). For military applications, sensors may have to be dropped under unfavorable operating conditions. Some innovative ideas about deployment are presented and discussed below. The common objective of all these ideas is to achieve optimum uniform coverage.

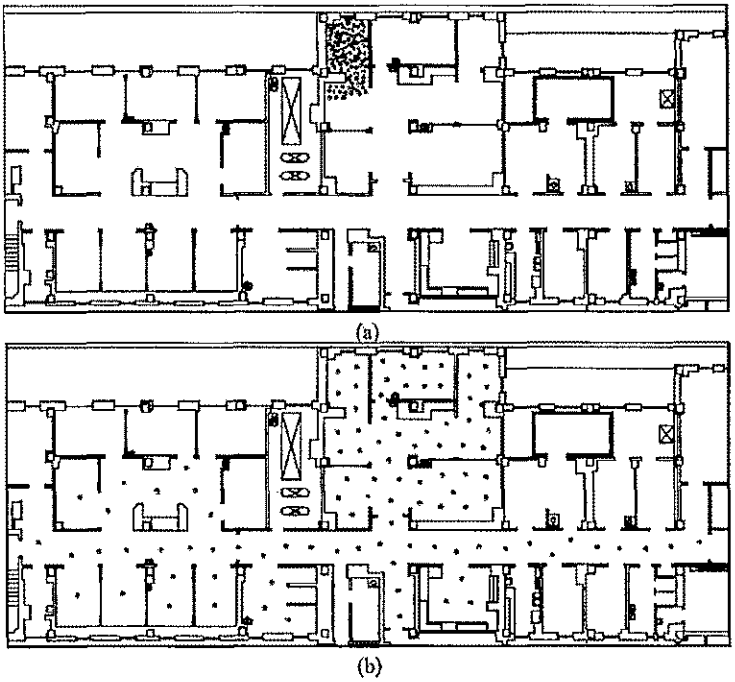
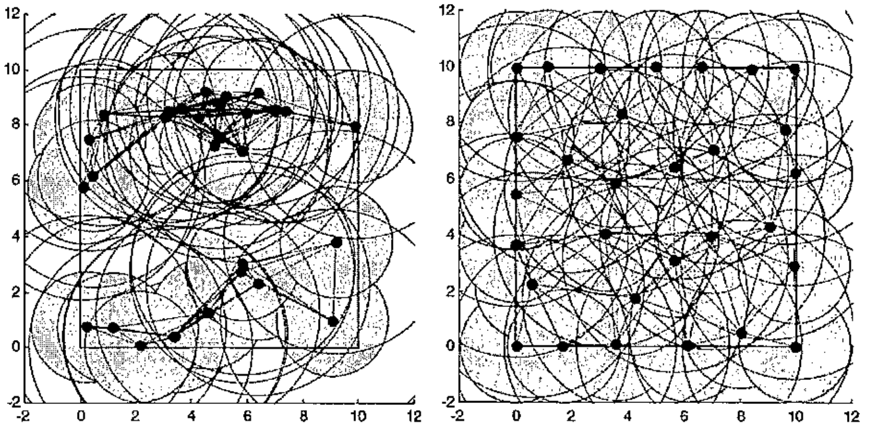

(a)
(b)

Fig. 6: Mobile Sensor Network Deployment using Potential Fields. A prototypical deployment experiment for a 100-node network. (a) Initial network configuration. (b) Final configuration after 300 seconds.

The paper by Andrew Howard et. al. [18] discusses mobile sensor network deployment. It describes a potential-field-based approach to deployment, in which nodes are treated as virtual particles, subject to virtual forces. These forces repel the nodes from each other and from obstacles, and ensure that an initial, compact configuration of nodes will quickly spread out to maximize the coverage area of the network as shown in Figure 6. The potential field approach described relies on only one assumption: that each node is equipped with a sensor that allows it to determine the range and bearing of both nearby nodes and obstacles. A paper on constrained coverage for mobile sensor networks by Gaurav S. Sukhatme et. al. [19] compares parameters such as speed of deployment and extent of coverage that can be achieved by random deployment with uniform dispersion and symmetrical, regular placement of sensors.

The paper by Nojeong Heo et. al.[20] presents an algorithm called Distributed Self-Spreading Algorithm to achieve deployment. The algorithm is inspired by



(a) Initial distribution of sensor nodes (b) Final distribution after running DSSA

Fig. 7: Distributed Self-Spreading Algorithm

the equilibrium of molecules, which minimizes molecular electronic energy and inter-nuclear repulsion. Each particle obtains its own lowest energy point in a distributed manner and its spacing from the other particles is almost the same. They introduce the concept of force to define the movement of nodes during the deployment process. The force is dependent on the distance between nodes and the current local density. Consider the pictures in figure 7. Tiny circles represent the positions of nodes and small (shaded) and large circles are used to show the sensing range and the communication range respectively.

Yi Zou et. al. [4] discuss the sensor deployment problem in the context of uncertainty in sensor locations subsequent to airdropping. Sensor deployment in such scenarios is inherently non-deterministic and there is a certain degree of randomness associated with the location of a sensor in the sensor field.

In some harsh environments, manually deploying sensors is impossible. Alternative methods may lead to imprecise placement resulting in coverage holes. Guilang et.al [6] proposes to deploy sensor networks composed of a mixture of mobile and static sensors in which mobile sensors can move from dense areas to sparse areas to improve the overall coverage.

CHAPTER 3

SENSOR CLOUD MODEL

Throughout our analysis, to aid understanding and simplify computations, certain assumptions about the physical construction of a sensor have been made, resulting in a simple design. Figure 8 shows a simple model of a sensor, an entity with a circle of influence defining a radius of communication and a radius of sensing. It is imperative to understand the size differences between sensor area and communication area. The sensor area need not be flush with that of the neighboring sensors while the communication area needs to. Obviously, that would result in gaps in the coverage of the sensor cloud. This is immaterial when one considers that a sensor cloud always operates in motion. More about this is discussed in 3.2.5.

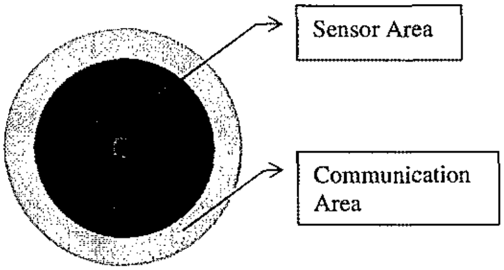


Fig. 8: Sensor Model.

Consider a sensor to be equipped with an omni directional antenna to communicate with its near neighbors. This paper does away with the gritty details of wireless communication for many reasons, some being that, the system proposed concerns itself only with the highest layer of any communication protocol used and that wireless communication between mobile entities is a subject in itself. Our assumption of the sensors possessing omni directional antennae will eventually decide the inter-unit distances in the sensor cloud.

A sensor cloud is a formation of mobile sensor units that holds itself together and moves around the environment as a single unit. There are no restrictions on the regularity between individual sensors. It builds itself together and sets off from a base station that either stays stationary or moves along with the cloud itself. Using mobile sensors, clouds of various sizes and shapes can be formed.

After the 'cloud' is formed it could translate or rotate. Analogous to building up, clouds could be broken down too.

The term "Sensor Cloud" refers to a collection of sensor-equipped robots that grows in numbers from a small size and navigates in the environment under study. There are some characteristics that define a cloud, such as connectivity among near neighbors at all times. However that is not to say that a cloud's shape is rigid. The same cloud may form different shapes to start with, depending on the deployment algorithm executed. The constituent sensors form a regular structure within the cloud. Deployment algorithms are entirely scalable. The same algorithm that formed a cloud of 100 sensors can form another cloud with the same form of 1000 sensors.

A sensor cloud's purpose is best served in tasks that require monitoring, detecting, classifying and locating specific events and tracking targets over a specific region of the environment, especially if the region is hostile or inaccessible. Since a cloud size is only restricted by the number of sensors, the cloud model works just as fine with a small field as with a gigantic field. Taking up a generic task, a sensor cloud model goes through a series of steps as part of the solution. The shape and size of the cloud to be formed is determined manually as per the situation and the objective of the operation. The cloud starts taking shape as per the rules of deployment. Upon completion the cloud sets off from the base and travels a prefixed trajectory or dynamically charts one. Over the course of its journey the cloud records the necessary observations and avoids obstacles. At the end of the journey, it returns to the base station and breaks down.

To present what this system has to offer in real world terms, analyze this. A vast tract of land has to be strip searched for a hazardous material. A sensor cloud takes shape and sets off from the base station and starts scanning the land row by row as the cathode ray gun does in television sets. Eventually it detects the element and calls the search off, retreating back to the base station. The location of the target material can be obtained either by setting up the cloud to beam the measurements off to a hovering aircraft or read it off to the personnel at the base station once it gets there. The exact position of the chemical agent can then be obtained taking into account the relative position of the cloud with respect to the base station and the relative position of the chemical within the could when it found it. Another real world problem lies in tracking moving objects. Since a cloud is mobile, it chooses the next direction to move depending on what part of the cloud last detected the object. One more situation where a cloud's adaptive ability would be brought into light is mine sweeping. In this case, instead of forming a regular shape, it lengthens itself and develops a linear shape optimized for the task.

This paper focuses on the fusion of robotics and wireless micro sensors. A huge number of mobile robots are straddled with case-specific sensors. Such a massive distributed network of sensors nodes requires robust, scalable, management techniques that provide a clean interface to every activity the sensor collection can possibly be imagined to be involved in. Developing a sensor network modeled on the cloud state is the topic of this thesis. This thesis envisages a system that involves navigating a swarm of mobile sensors, what we refer to as a "Mobile Sensor Cloud", across large swathes of terra firma.

This thesis classifies collections of sensors into 4 types depending upon the deployment method employed and connectivity among neighboring sensors.

gaseous: A collection of sensors that when deployed spread further apart from each other occupying the entire volume of available space. Though the possibility of the final state being a continued motion state such as the brownian motion is not improbable. This classification includes the work of [18], [19], and [20].

dispersion: In some situations, sensors are air-dropped or dispersed through some means over the field of interest. Since the final placement of the sensors cannot be accurately controlled, it results in an uneven distribution as shown in [4]. Techniques like virtual force algorithm (VFA) can then be used to insure uniform monitoring of the field[21].

cloud: The cloud state is analogous to the fluid state in the three physical forms of matter. However unlike fluids the constituent nodes maintain a uniform geometric arrangement like that found in a crystal. It is marked by the presence of cohesiveness among its constituents. The cloud can navigate to any corner of the available space, but the whole cloud moves as one unit.

static: One of the most common and popular arrangement of sensors is the static one. Sensors can be manually attached to ceilings or other fixtures. Several models have been proposed [22].

The sensor cloud is the centerpiece of this thesis. This chapter discusses the sensor cloud model, its capabilities and the different characteristics of members of its own kind. True to its literal interpretation a sensor cloud is a coherent mass of sensors that takes shape from a few sensors and floats or wanders around in an environment. Figure 9 is an abstract representation of a sensor cloud that serves its purpose in understanding a cloud's form. The sensor cloud model is comprised of the physical cloud itself with its constituent sensors and its management.

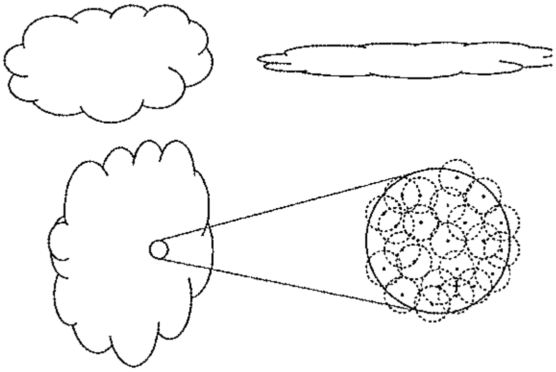


Fig. 9: Sensor Cloud Model.

3.1 CLOUD MANAGEMENT

Figure 10 is a representation of one of the many uses of a sensor cloud. A cloud can change shape either with the existing number of sensors or by putting on a few sensors more, knows precisely where each of its individual sensors is located within itself, translates from point A to B, and rotates about itself. Cloud management comprises all tasks that are required of a cloud to finish off a task. It can subdivided into

- Mobility Management
- Localization
- Data Management
- Control Management
- Power Management

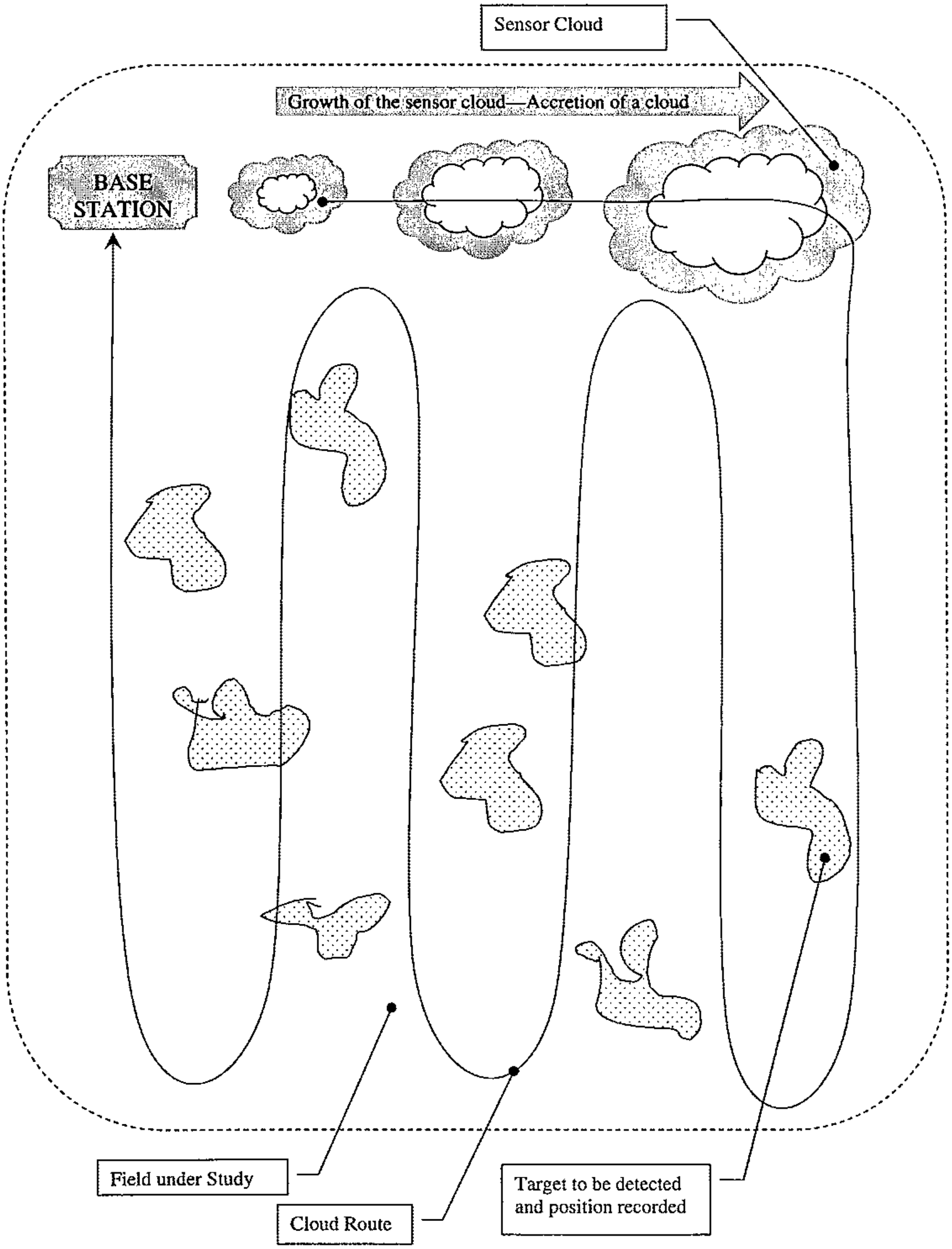


Fig. 10: Mobility Management.

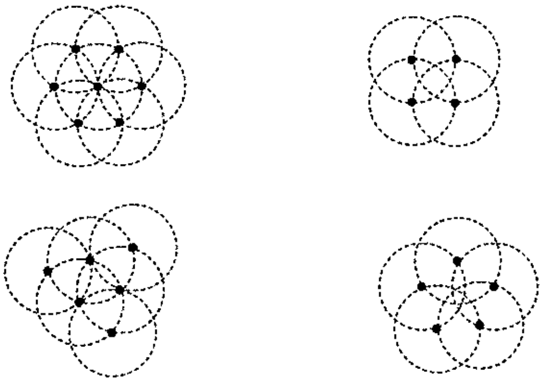


Fig. 11: Samples Initial Patterns.

3.1.1 Mobility Management

Deployment

Deployment concerns itself with the initial pattern of sensors, how they group together, how far apart they are from each other, or how the pattern magnifies itself, in case it decides to do so. The following tasks make up deployment.

- *Initial Pattern:* To start with, a group of sensors should form a unit pattern such that it is both simple to be manually placed and easy to set off the chain process to form the cloud. In other words, sensor units must be able to arrange themselves into an acceptable pattern starting from an initial arrangement that is easy to realize in a convenient deployment scheme. The deployment strategy varies with the issue at hand. In case of adequately known environments, sensors can be strategically placed. In case of the environment being hostile or unknown, sensors need to be deployed by other means. The effectiveness of a mobile sensor web is determined to a large extent by the coverage provided by the sensor deployment. The positioning of sensors affects coverage, communication cost, and resource management. Figure 11 shows some sample initial patterns of sensors that will either form a mobile cloud in itself or will grow to form bigger ones.

- *Accretion of a Cloud:* A. Howard et. al. have described a similar process of letting loose a bunch of sensors to cover a surface and to maximise the coverage of the sensor network thus formed[18]. A pattern made up of

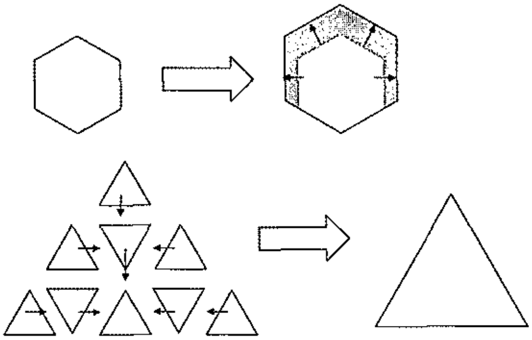


Fig. 12: Accretion of a Cloud.

a few sensors is too small to be considered for any significant real world missions. The coverage which eventually determines the sweep area is negligibly small. That is why the pattern needs to build upon itself and form a larger cloud. A larger cloud makes wide strides while lumbering along and intercommunication between sensors is what keeps the shape from falling apart. There exist many ways to grow a smaller cloud into larger one. In addition to this, the ability to interlock with peers is another deciding factor. Triangles, squares and hexagons can interlock with another of their own kind with ease, while pentagons cannot. Factors that impact performance are simplicity of development and growth per unit time. Growth may be brought about by coalescing siblings or by building the parent cloud itself a couple of sensors at a time as shown in Figure 12. Moreover, using accretion, a cluster of sensors can be created at one point and set off into the field for monitoring and other purposes.

Yi Zou et. al. deals with air dropping of sensors onto a battlefield or a wild life preserve to provide a continuous coverage of the environment[4]. However, access to every location of interest is either limited or dangerous. In these situations, a cluster of mobile sensors can be formed in a safe location positioned on the periphery of the field and can be set off into the region. Hence, in events of restricted access either circumstantial or natural, the ability to build up and break down a cloud pays off.

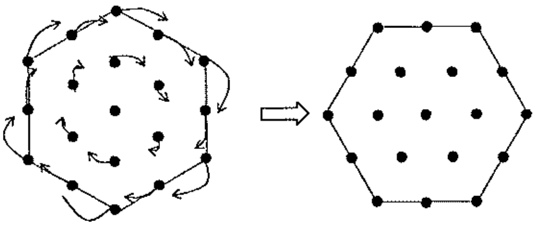


Fig. 13: Rotation of a Cloud.

Navigation

Navigation encompasses all processes that are brought about either by translation or rotation. The objective put forth, the type of cloud to work with and the environment determine the kind of navigation to be undertaken. In straight terms, navigation can be described as moving from point A to a point B. Performance parameters for navigation include speed, computational overhead, reliability and recovery in face of sensors turning up dead or rogue, and obstacle tackling. Let us discuss the different aspects of navigation.

- *Translation* During translation, a cloud may either move all sensors, or a few sensors at a time. Moving all concurrently may result in faster locomotion but the speed advantage may be offset by the complexity of computation involved and moving few sensors at a time does the exact opposite. As with any other kind of motion, a sensor cloud needs to retain its shape at the end of translation. Translation is possible only in those directions that the degrees of motion permit. Sometimes the cloud needs to re-orient itself through rotation to translate in a particular direction.

- *Rotation* Sometimes a cloud may have to rotate to align itself in a new direction. A hexagonal cloud may need to readjust itself to point in a different direction as shown in Figure 13. The need to realign itself may arise for different reasons. It may be so that realigning itself helps in avoiding an obstacle which was earlier blocking its path. It may also be such that the shortest path to a destination falls on a slower degree of motion. Moreover it's not unlikely that moving along some axes of motion is faster than some others.

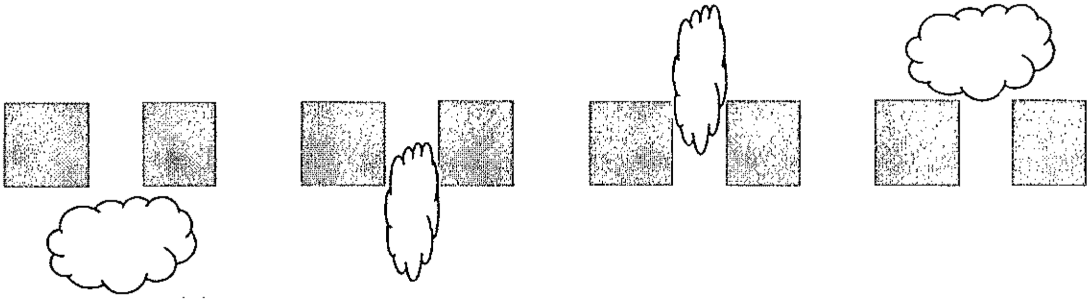


Fig. 14: Passing through a passage.

Obstacle Avoiding

A cloud's ability to retain its shape during motion may be challenged by obstacles blocking the path. The course of action upon confronting an obstacle depends on the shapes and sizes of both the cloud and the obstacle. Obstacles may result in invalidating one or more of the axes of motion. Also, the sensors on the periphery are the ones to detect obstacles first. Hence it makes sense to equip those sensors with an additional module of obstacle detection in order to avoid colliding with the hurdle. Figure 14 and Figure 15 show a pictorial demonstration of the how a sensor cloud may react to obstacles. Figure 14 reveals one way of obstacle tackling where a sensor cloud decides to and is able to squeeze itself through the narrow space.

Recovery

Recovery refers to the process of the sensor cloud, returning back to the base station in the event of a severe failure. Failure for a sensor cloud can be defined as the loss of any of the mobility, communications, sensing, or data processing modules of the sensors. If the sensor cloud finds itself incapacitated by the failure and goal achievement seems highly improbable, then a group of sensors decides to tell the whole cloud to head back home. The responsibility to make a decision to retreat may either fall on a group of sensors predetermined as the control unit of the cloud or an ad-hoc group. Whenever a sensor finds one of its key instruments faltering, it may send a distress signal to its neighbors. As more and more sensors start decaying, the distress signals may rise in number eventually leading to the control group deciding to veer off its course and retreating. The control

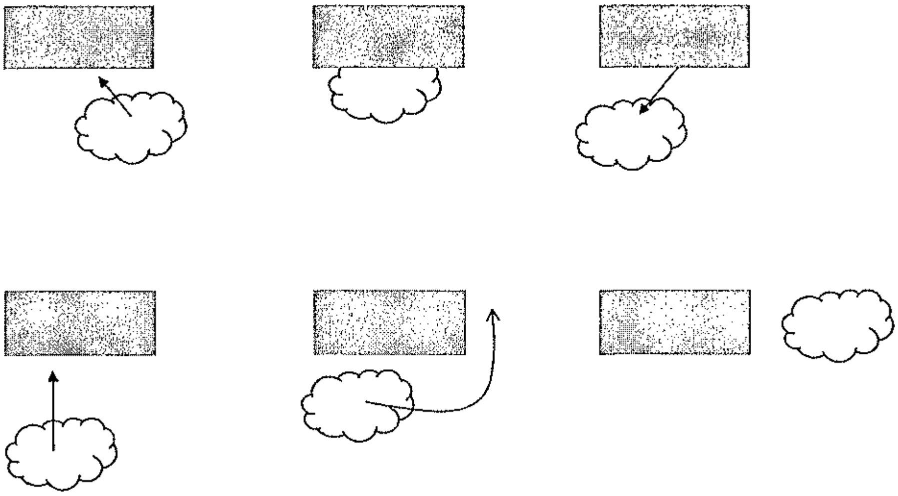


Fig. 15: Deflecting/Sliding Off an Obstacle.

group makes use of pre-determined performance grade to decide the whether the prevailing circumstances warranty recovery over navigation. As an example, consider a sensor cloud on a reconnaissance mission. Performance parameter for such an objective can be quantitatively defined as the ratio of the fraction of the total distance traversed and the fraction of the total number of sensors that have turned ineffective. If the performance parameter dips below a certain value, then it makes better sense to call off the mission and bring home the sensor cloud. In fact, recovery is almost a function of the fault tolerance of the cloud.

3.1.2 Co-ordinate systems

Every sensor in a cloud needs to be assigned some sort of an identification tag that is unique to itself. A co-ordinate system needs to be set up to locate each sensor. The shape often determines the simplest coordinated system that can be chosen. For example, a regular cartesian co-ordinate system is suitable for the sensor cloud in Figure 17, a cartesian system with oblique axes for Figure 16 and a polar co-ordinate system best fits for the one in Figure 18. Further, the environment that the cloud traverses might be referenced with a set of coordinate systems different from the one used within the sensor cloud. The co-ordinate system could either change or stay static during motion. Thus if sensors in the cloud retain records of their movements and orientations at all times, it is possible

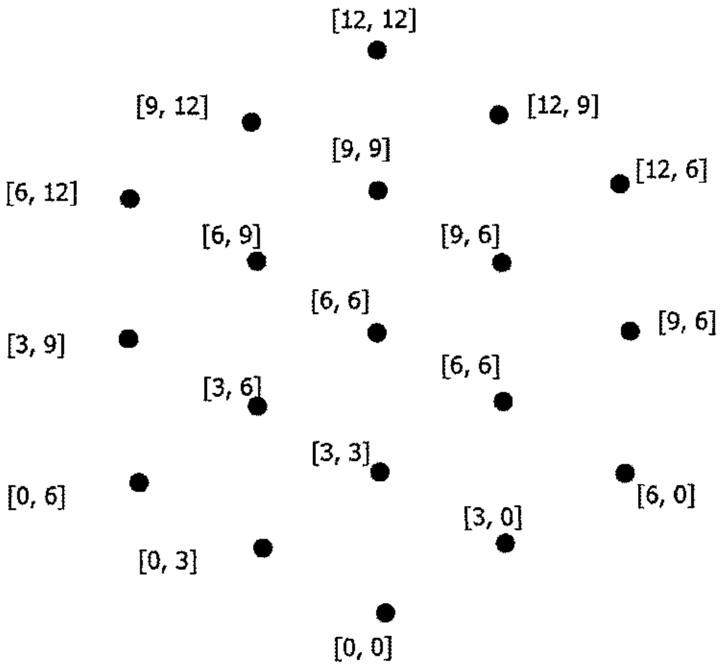


Fig. 16: Co-ordinates in a Hexagonal Cloud.

to compute their global coordinates.

3.1.3 Control Management

Control of a cloud refers to the course of actions the cloud must take when confronted with obstacles, and failing sensors. Depending on the share of the responsibility taken up by the cloud and the base, three different modes of control can be stated.

- Centralized: Usually a cloud along with the base station is monolithic. They both work in tandem, the sensor cloud feeding back collected data back to the base while the base passes on instructions to the base on what to do next. Hence, if the base is mobile, it is within the communication distance of the cloud, which relieves the cloud of storing the data gathered.
- De-Centralized: However, in certain conditions the base may not find it feasible due to terrain difficulties to trail behind the cloud, or at least within the communication range of the cloud. In these situations, the cloud and the base are two separate autonomous entities. The cloud now has to handle

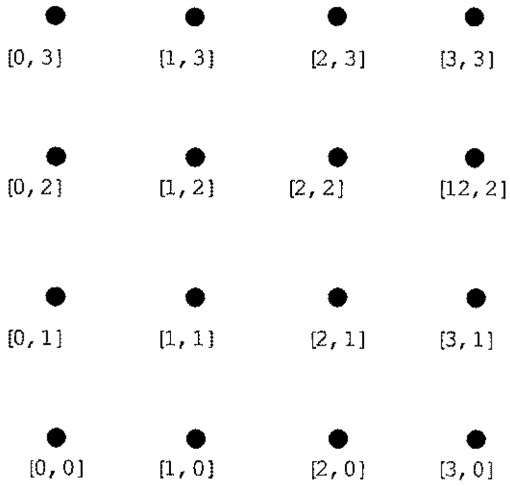


Fig. 17: Co-ordinates in a Square Cloud.

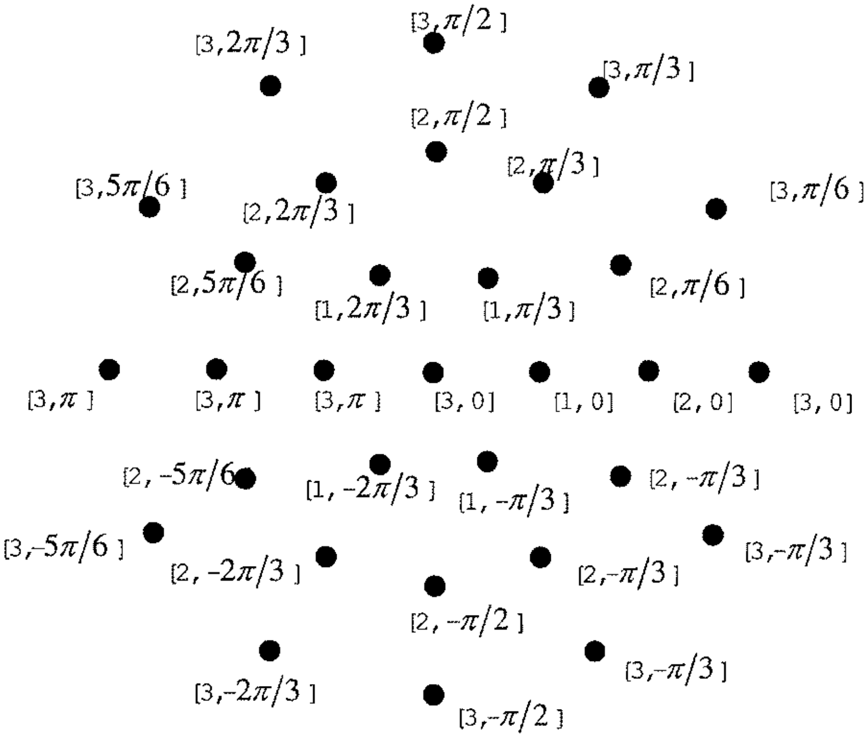


Fig. 18: Co-ordinates in a Circular Cloud.

data management additionally. The sensor observations now have to be stored in the sensor memories and if space is found inadequate, compression techniques have to be used.

- Mixed: As the name implies, mixed mode offers distributed ownership of the cloud and its activities. Decisions that require frequent data exchanges between the cloud and the base, and that are time restricted will be made at the cloud. Long term decisions that take time and need human interaction are transferred to the base.

3.1.4 Power Management

For static arrangements of sensors, power conservation is a serious matter. When clouds are considered to be mobile, it implies they have enough power reserves to drive themselves, and run other low-power modules such as the communication, sensing, and computation modules. However technological advancements in drive mechanisms in the near future may result in systems that really run on low power. They may be on the same scale as the power requirements of computation modules. In these situations, it seems imperative to have some sort of power management in the cloud model. That leads one to think that power is one resource that cannot be taken for granted and the whole survival of the sensor cloud operation hinges on effective power management.

A sensor cloud needs to conserve its power to run longer. Dynamics energy saving techniques like shutting down power hungry devices hold promise[23]. However it is known that of all the instruments hitched on to a sensor, communication modules are the ones that are power guzzlers. And a sensor cannot afford to shut itself off from its neighbors and thus become disconnected. To get around that, modules like the sensing module, the data processing module, anything other than the communications' module can enter a low power mode and go to sleep. The sole working communications model in turn opens up a wake-up channel specifically reserved for receiving wake up calls from their fully turned on peers. This wake up call is transmitted only when specified conditions are met. These conditions may vary as per the measure of effectiveness requirements. These peers are either distributed across the cloud in patterns that are decided by the cloud shape or are the ones on the periphery of the cloud. In return for the power saved, the cost to be paid lies in the penalty for missed or overlooked activity-prone or affected areas. Communication power usage is kept low by only requiring communication with near neighbors.

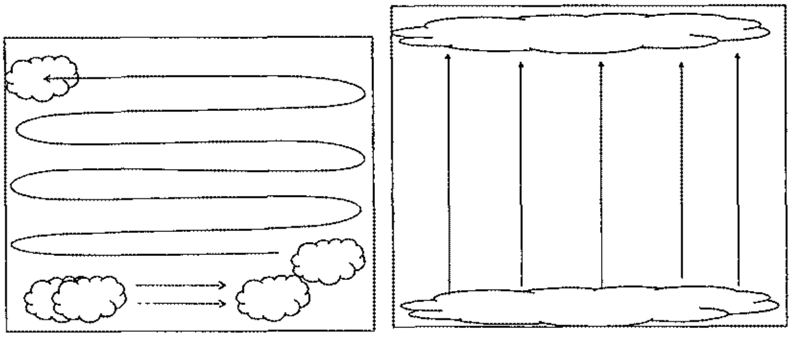
Though power management is critical to the mission, it cannot surpass the mission itself in relevance. Some examples of mission objectives are sniffing out chemical agents or shadowing fleeting entities of some sort. The one action that is independent of the mission type and which is critical to its success is that of scrutinizing the ground for something. Hence there is a trade off between the measure of effectiveness of the scrutinizing action and that of the power management. Performance metrics can be defined as following: a power saving of 30% or above can be achieved at the cost of not detecting targets smaller than 3 sq.ft across. For targets that are temporal in nature or are shifting in position, performance factors may vary. This would mean that a cloud that has partially shutdown some of its sensors would need to travel at a speed slower than when all its sensors are turned on fully. Mobility of the cloud helps in offsetting the losses in the measure of effectiveness. Patterns of the sleeping sensors can be so arranged that no piece of ground escapes from being frisked over by the 'awake' sensors.

3.1.5 Sweeps

A sensor cloud provides different types of coverage, like a carpet sweep coverage or a barrier coverage[14]. As Figure 19(a) shows, a sensor cloud can be used to scan an environment for some specific element(s). Also, Figure 19(b) shows yet another way of doing the same but from a different perspective. In this case, the same cloud changes its shape into a slimmer one that has a larger width. It's called a barrier and it acts like a sentry, impermeable for mobile foreign agents. However, if the barrier can move forward, it is said to provide a barrier coverage and now it can stop any foreign element from passing through.

3.2 CLOUD CHARACTERISTICS

A cloud's characteristics are determined by its size, shape and the actions it performs.



(a) Carpet Sweep (b) Barrier Sweep

Fig. 19: Sweeps

3.2.1 Connectivity

A cloud is a fully connected graph of nodes, i.e., there is a path from any point to any other point in the graph. Nodes in the graph are mobile sensors and edges are defined by near neighbor communication. The vertex degree associated with each node may change while carrying out manoeurves. Further, it might start as a complete directed graph but some of its edges could turn unidirectional, owing to sensor communication failures.

3.2.2 Density

Apart from the different directions of movement, one more factor that stands out is the number of neighbors a sensor has. The more neighbors that a sensor is in touch with, the better the chances of retaining the shape of the cloud while in motion. This is based on the fact that a cloud's constituent sensors are more likely to land further from their intended positions during motion if there are fewer static neighbors to correspond to and cross check their bearing. Moreover a sensor is better able to localize itself in the face of more reference sensors. As evident from Figure 11, the hexagonal cloud has 6 neighbors per unit, the square one has 4, the triangular one has 6. Maintaining synchronism is one critical element necessary for the functioning of the sensor cloud. Inter-sensor distances is a parameter that is decided by the sensor reading granularity required of the operation.

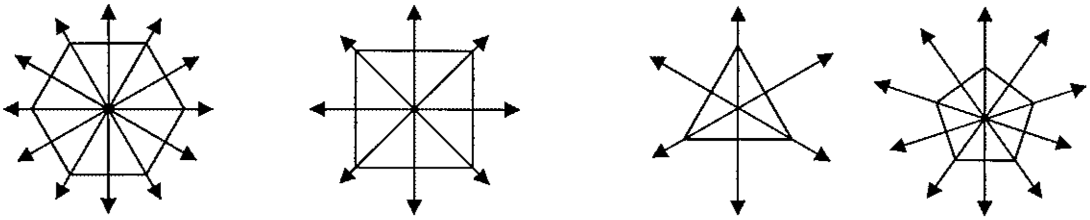


Fig. 20: Degrees of Motion.

3.2.3 Degrees of Motion

Degrees of motion are defined based on ease of movement. Figure 20 shows convenient degrees of motion for several configurations. Analyzing the four patterns in Figure 20 we can see that the hexagonal one has 12 directions of motion while the square one has 8, the triangular one has 6 and the pentagonal one has 10. Theoretically, any of these patterns can move in an any number of directions, but we concern ourselves with only those that offer the least computational overhead. However, asymmetrical axes of motion are entirely possible, the only limiting factor being that the computational overhead involved could overrun any benefits to be realized.

3.2.4 Cloud malleability

A cloud's ability to change shape as per the surroundings is critical during navigating around obstacles. As Figure 14 demonstrates, a sensor cloud does need to change as per the conditions.

3.2.5 Coverage

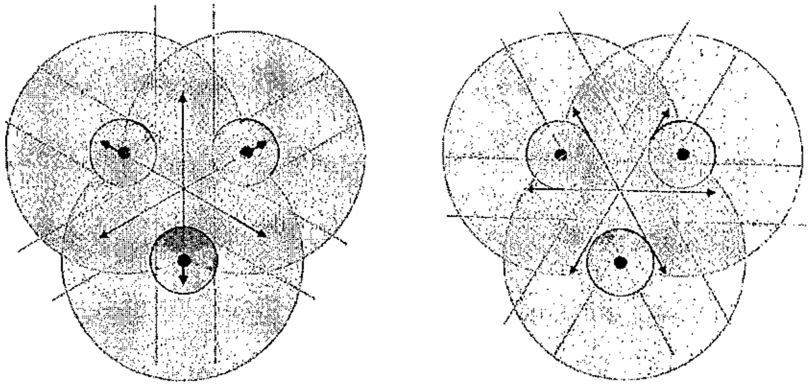
Coverage stands for the effective area swept by a cloud not stationary but while in motion. The point here is that even though a sensor cloud may have some gaps that lay uncovered due to spacing between sensors, that does not matter because the cloud, for a greater part of time, is always in motion. So, necessary attention has to paid to see that a sensor cloud sweeps every inch of ground in its path while moving. To what extent does a cloud perform is largely decided by its shape and how it moves, which could either be linear or rotatory. To recognize whether a cloud can provide complete coverage in its axes of motion, it suffices to prove that no straight line exists that cannot pass through the cloud without intersecting the sensor area of any sensor as suggested in the left picture of Figure 21.

However, a cloud cannot move in such axes of motion that allow intervening gaps unswept. Axes of motion that result in sweeping the gaps are termed valid, otherwise they are called invalid. As the picture on the right in Figure 21 suggests, the resulting sweep would reveal gaping holes. Moreover, the valid axes of motion is directly related to the cloud shape and inner arrangement of sensors.

Observe the sensors shown in the upper picture in Figure 22 with communication range R, and sensing range r. It is fairly obvious that inter-sensor distance d cannot be greater than R. To determine the maximum $d \leq R$, consider the lower picture in Figure 22. To avoid any uncovered gaps while moving, the maximum height of the triangle should be $2r$. Hence,

$$\begin{aligned} d\cos 30^\circ &= 2r \\ d &= \frac{4r}{\sqrt{3}} \end{aligned} \tag{1}$$

Thus, if they are to move in any direction, d should be $min(\frac{4r}{\sqrt{3}}, R)$.



(a) Valid Axes

(b) Invalid Axes

Fig. 21: Valid/Invalid Axes of Motion

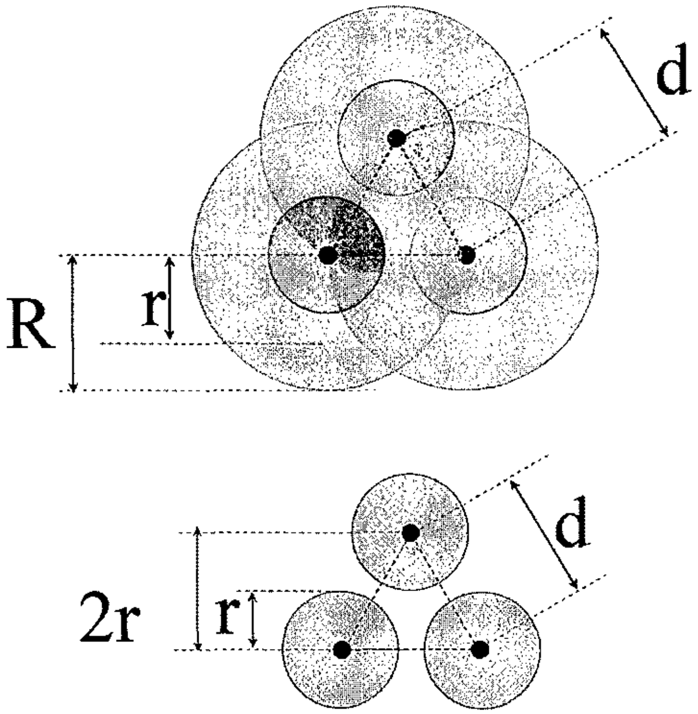


Fig. 22: Minimum inter sensor distance

3.2.6 Fault Tolerance

Fault tolerance refers to the resilience of the functioning and the accuracy of the data recovered by the cloud in the event of sensors dying or some of the key instruments of sensors wavering in their accuracy. Hence as such, fault tolerance can be studied with reference to either cloud integrity or data integrity.

- Cloud Integrity: It can also be quantitatively defined as the maximum number of entities that can be removed from the cloud and still maintain cloud integrity. Faults may either occur in communication modules, sensing modules or transportation modules. The degree of decay caused by a fault directly impacts the ability of a cloud to retain its performance index.

- Data Integrity: It refers to fault tolerance of a sensor cloud system in preserving the fidelity of the sensor observations. Data harvesting can be done when the sensor cloud returns to the home base. And further, the data can be processed to weed out the errors. Several ideas have been put forth to extract relevant data from faulty observations. The technique of functional characterization of integration of abstract interval estimates of sensor readings by Prasad, L et. al is one means to achieve the objective [24]. However, fault tolerance can be put ahead in the process to the stage when the sensor readings are being recorded. Krishnamachari, B and Iyengar, S have dealt with feature extraction in Distributed Wireless Sensor Networks [25] [26].

Moreover, a cloud is capable of graceful degradation. When a few sensors go off, a cloud reorganizes itself in different ways. One way would be to rearrange itself such that the now remaining sensors are placed farther apart. The cloud retains its shape but becomes less dense. Another way would be to retain the density and shrink itself. However, the cloud as a unit continues to function in event of hardware failures or software errors with only a expected decrease in performance. A discussion on sensor relocation in mobile sensor networks by Guiling Wang et al considers among others the concept of moving previously deployed sensors to overcome the failure of other nodes [27]. In conclusion, mobility enhances the fault tolerant capability of a sensor cloud.

CHAPTER 4

A HEXAGON SENSOR MODEL

This chapter discusses laying out a hexagon shaped sensor cloud based on triangular building blocks and rules for deployment and navigation associated with it. Two specific implementations will be explained regarding building up the cloud.

4.1 CLOUD DEPLOYMENT

Cloud deployment is an important task in the cloud model. It relates to the formation of the cloud from an initial pattern of sensor nodes. The type of deployment depends on the sweep required of the cloud and the overhead that can be handled during cloud formation. The overhead involved is usually a trade off for the speed and resulting shape of the cloud formation. This chapter introduces two methods of deployment. Order n of a hexagonal cloud is the number of sensor nodes on any of the sides of the hexagon. To form a hexagon of order n, $3n^2 - 3n + 1$ sensor nodes are required. The width of the swept area when a cloud translates linearly depends on its orientation. In the case of a hexagon, the maximum sweep results during movement in any of the directions along the diagonals, which for a n order hexagon is $2[(n-1)d + r]$, with d being the inter-sensor distance and r being sensor range respectively. At this point, two different methods to start building up the cloud will be discussed. The two methods differ in the way the mobile sensors route their paths to their destinations and the messages that transpire between them and their steady counterparts.

4.1.1 Flag Technique

This technique starts building up from an initial pattern of sensors as shown in Figure 23. The initial pattern is simple to manually lay out and uses just 4 sensor nodes irrespective of the size of the cloud to be formed. These initial sensor nodes are to be placed manually and hence are called prepositioned sensors. Subsequent sensors called self-deployable sensors are placed within the primordial cloud as indicated in Figure 24.

A regular hexagon cloud consists of three kinds of sensors, the core or C sensors, the trunk or T sensors and the foliage or F sensors as shown in Figure 25.

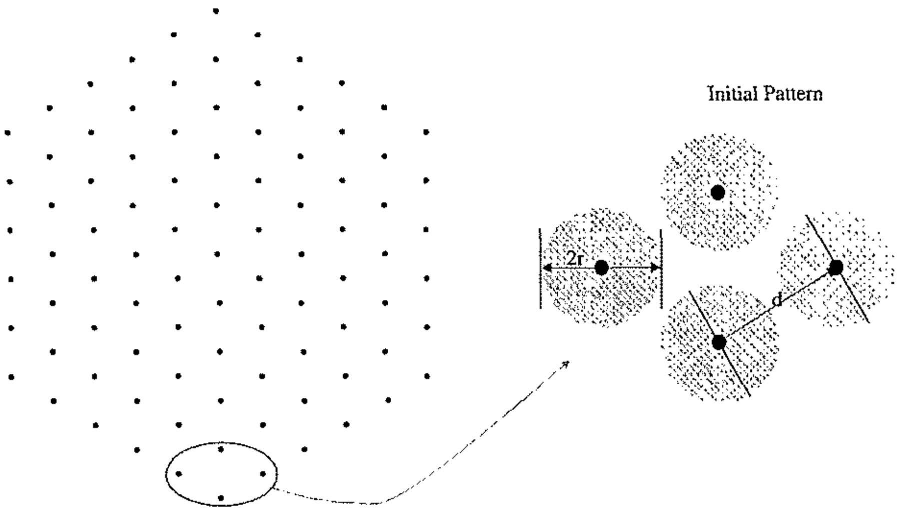


Fig. 23: Hexagon Cloud.

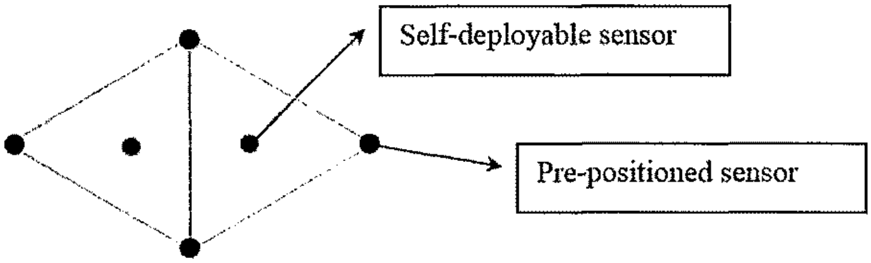


Fig. 24: Initial Pattern.

This differentiation is only on the basis of their positions; their functionality is identical. The T sensor columns are the 2 columns on either side of the C sensors at the center of the cloud. The F sensors are the surrounding columns on either side. The initial pattern shown in Figure 23 is the foot of the T sensor column. Every cloud first begins with the construction of the C and T sensor columns followed by that of the F sensor columns.

This method of deployment is called so because it involves transferring flags between sensors to build up the cloud. This method allows for building the sensor cloud in two ways.

First build the entire trunk in one go and then start on the foliage or build

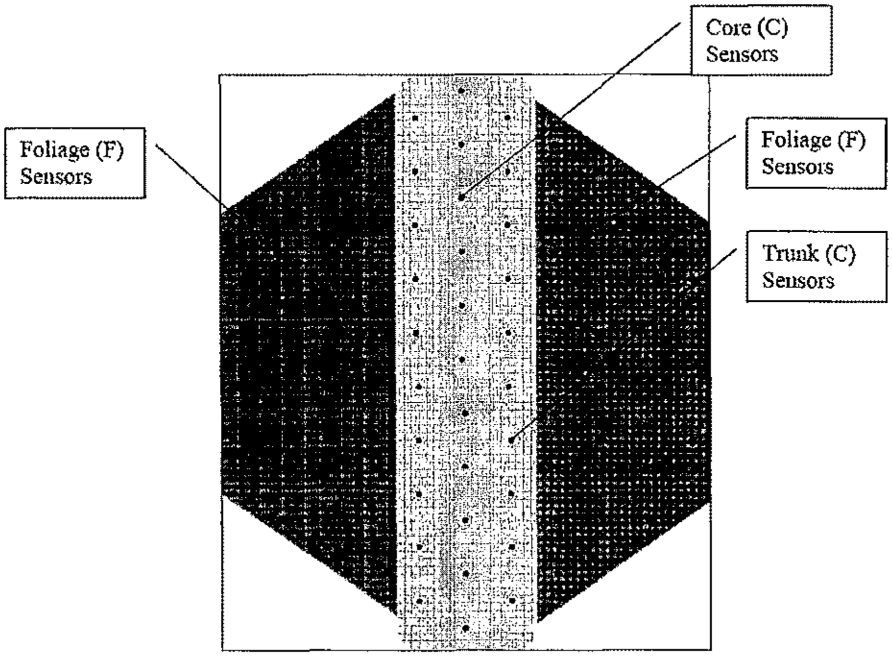


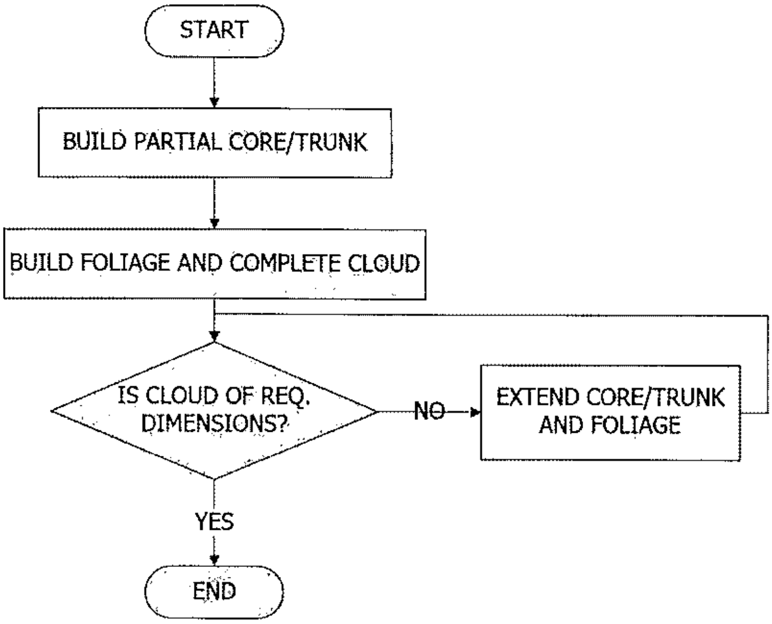
Fig. 25: Core, Trunk and Foliage sensors.

trunk and foliage in bits and pieces one following the other, the trunk starting first. The two methods are represented in Figure 26 and illustrated in Figures 27 and 28. The second method requires knowledge of the number of sensors ready at hand prior to cloud formation. C, T sensors and F sensors differ in the way they move during the deployment phase.

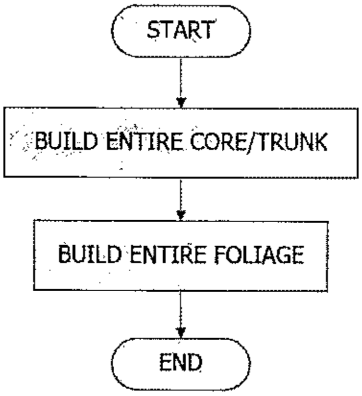
Actions involved during cloud formation can be categorized into low-level and high-level actions. Low level actions, as the name describes relate to individual sensor movements. These actions pertain at the sensor level and consist of generating and parsing messages transpiring between the moving sensor and its closest steady senors. High level actions follow Core, Trunk and Foliage formations. Hence to bring about a single high level action, a series of small low level manoueuvres have to be carried out. The following sections describe low level and high level actions.

Low Level Actions

Self deployable sensors are either at position within a triad of sensors or moving through a pair of sensors. Self deployable sensors follow the process as illustrated



(a) Partial Trunk and Foliage



(b) Complete Trunk and Foliage

Fig. 26: Different sequences of cloud build up.

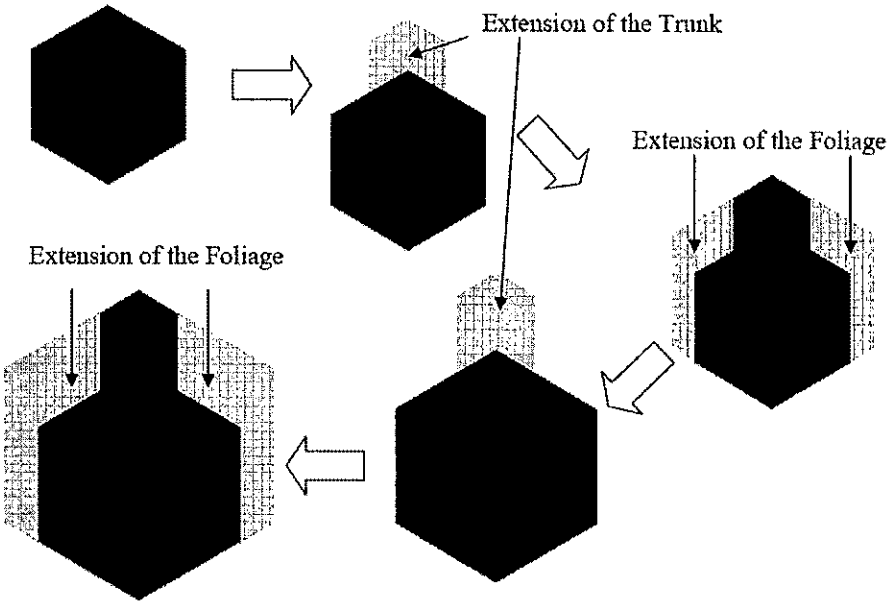


Fig. 27: Complete Trunk and Foliage.

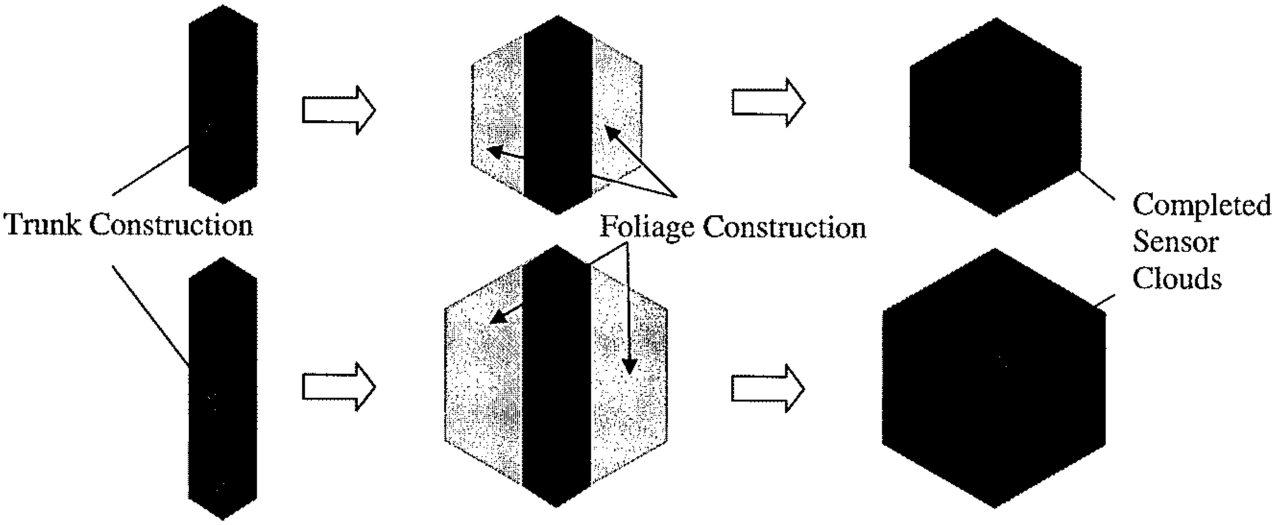


Fig. 28: Partial Trunk and Foliage.

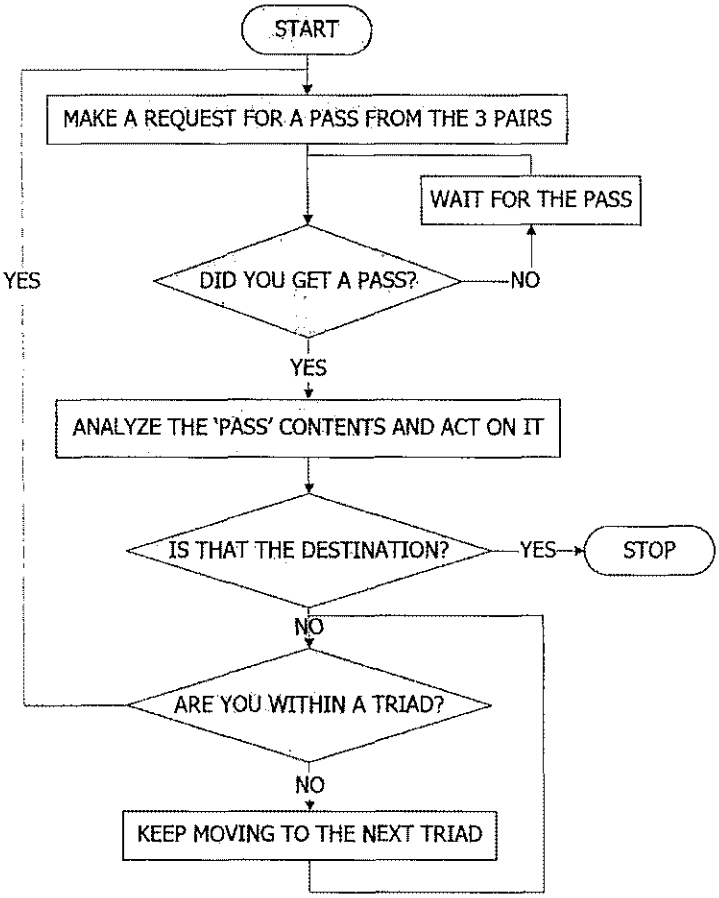


Fig. 29: Decision Flow for a Self Deployable Sensor.

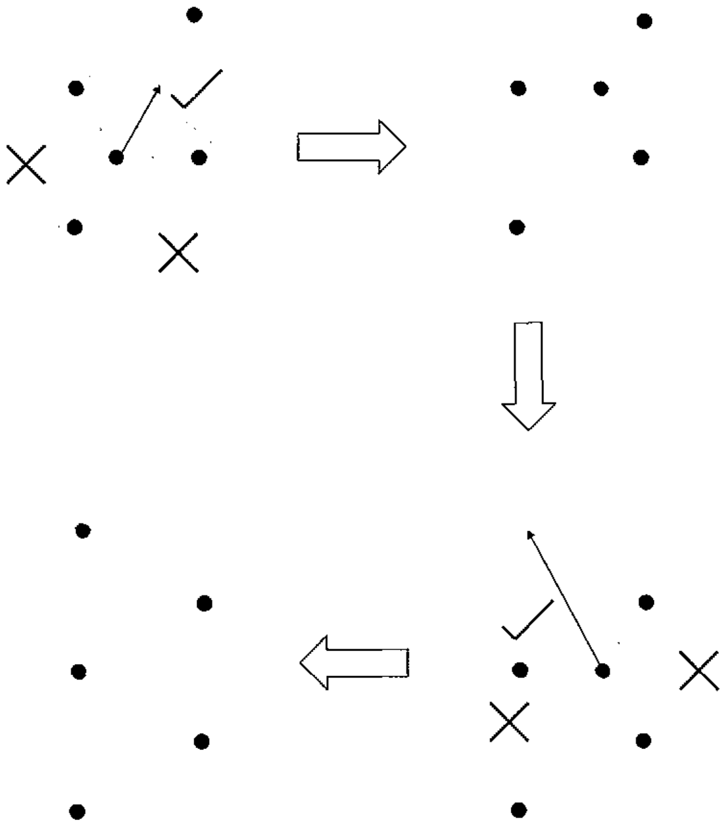


Fig. 30: Ask and Pass process.

in Figure 29. The same process is further illustrated in figure 30. The gray sensor node represents the child node and the black ones are the parent nodes. A child node always starts within a triad. Assuming that the pass for moving on is passed on by the upper pair of parent nodes enclosed in the gray oval. A child node then starts moving in between the two nodes always being equidistant from them. It goes on until it has reached the center of the upper triangle. Had there been no sensor node at the vertex of the upper triangle, the child node would have proceeded on to occupy the upper vertex position. Upon sensing the presence of the parent node at the vertex it falls into waiting phase. Assuming that the upper left pair of nodes nods in approval, the child node proceeds to the center of the triangle. After reaching over there and not finding any new parent at the vertex of the new triangle, it goes ahead and occupies the vertex position. A child node knows it has reached the vertex position by its distance from the two parent nodes that it set out from.

High Level Actions

A self deployable sensor keeps moving from one triad to another endlessly unless it comes to a triad in which one of the three pairs decides to let in that sensor to form a third sensor in an adjacent partially formed triad. In other words, the pair needs to send a message to the self deployable sensor to fill in a vacant steady state spot. However, a pair can only issue such a message if either of it is in possession of a flag, called the 'go-ahead' flag. The following paragraphs describe the construction of the three segments of a cloud, the core, trunk and the foliage. The transfer of the 'go-ahead' flag between sensors plays a significant role.

Formation of the Core and Trunk

It follows from intuition that motion of the C and T sensors is always restricted to directions at a fixed angle to either of the axes. That makes the paths of T-sensors and C-sensors simple to route and require no complex computations. Observing the picture in figure 31, one can notice that once a T-sensor is launched from the base it can charter its way out to its destination through a series of alternate right and left turns.

Let us study the path traversed by one such T-sensor. Consider figure 31. To start with B0 on the left side holds the 'go-ahead' flag. The first two self deployable sensors set off and occupy the B1 spots on either side of the cloud. Every two T sensors is followed by a C sensor. The first C sensor occupies A2. This process continues until the core and trunk reach the desired dimensions. Another T-sensor starting from S makes it to B2 as it can neither pass through (A1, A0) or (A0, B0). After positioning itself, B1 receives the flag from B0. After each pair of T-sensors is set off, it should be followed by a lone C-sensor, set off on any of the two sides of the quartet. This lone C-sensor is assigned to position itself on the center column of the cloud. Further, as more T sensors are set off on the right side, we have positions B2, B3 taken up. The last C-sensor treads its way to A4. B3 retains the flag. This flag can only be passed now to another T-sensor as it passes by, both lying on the same column, or to a F-sensor that is let in sideways. The core and the trunk can be extended as long as T-sensors and C-sensors are set off respectively at the starting point.

Formation of the Foliage

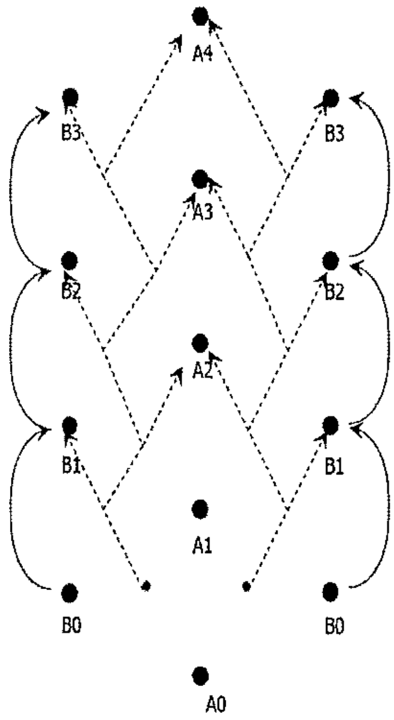


Fig. 31: Trunk Formation.

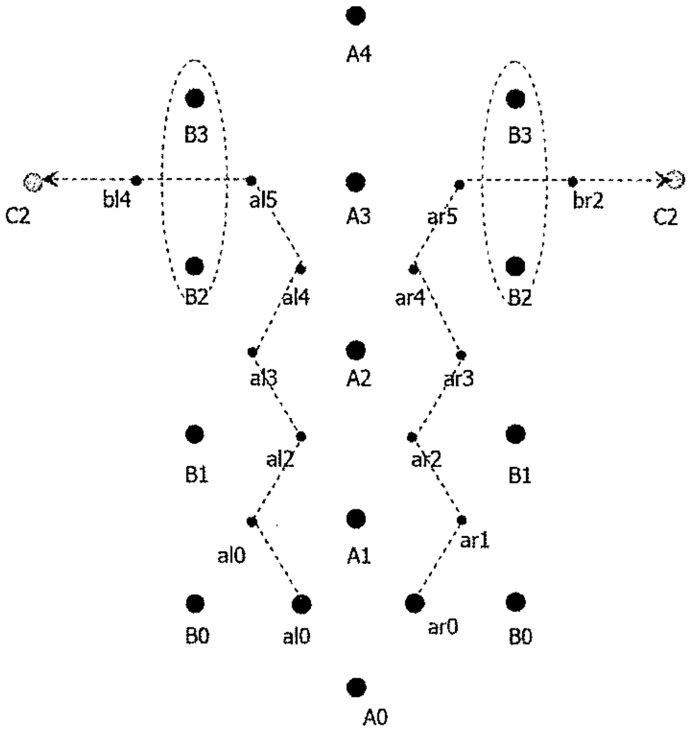


Fig. 32: Foliage Formation, a.

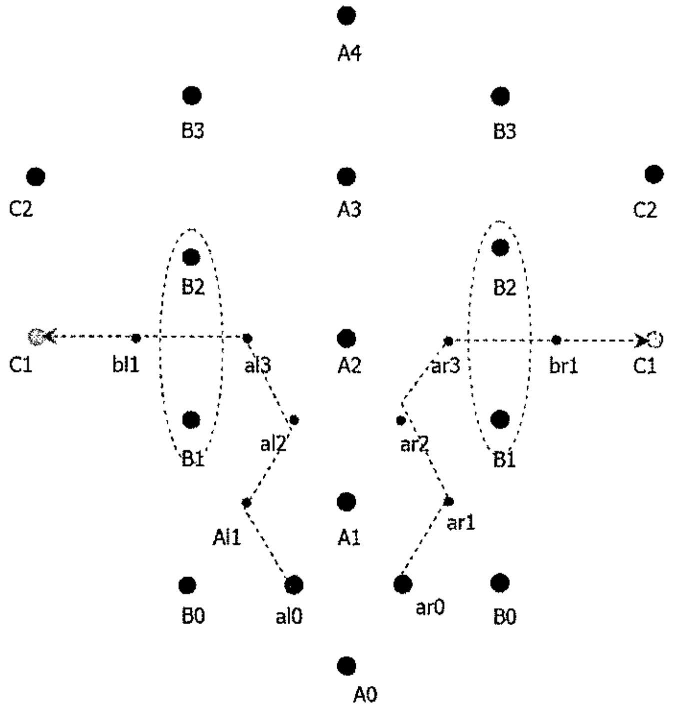


Fig. 33: Foliage Formation, b.

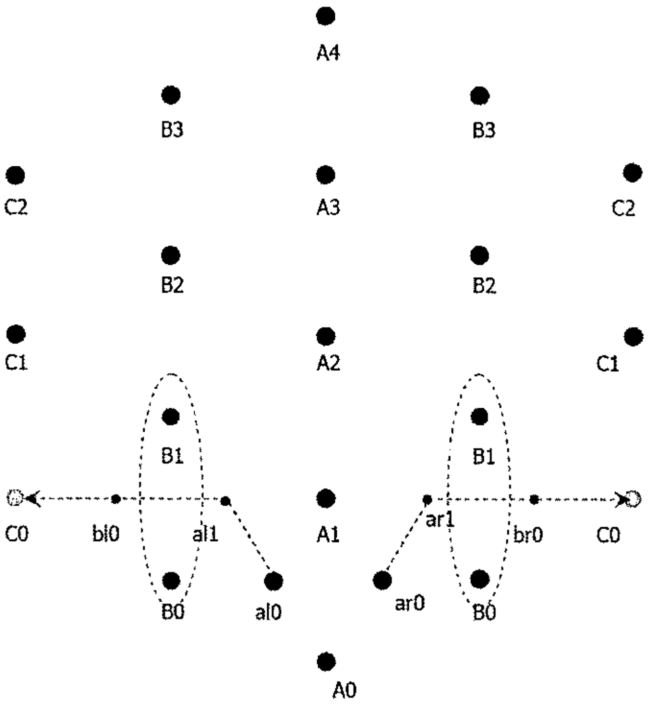


Fig. 34: Foliage Formation, c.

The trunk formation lasts as long as it is required for the hexagon of that degree. In other words, a hexagon of side length 3 sensors wide needs a trunk to extend to the A4 position. Consider the construction of the hexagon shown in figures 32, 33 and 34. Assuming the trunk has been built we will proceed to the construction of the sensors on the 3rd columns on either side of the trunk. Following are the basic principles for determining the route of a F-sensor. The path of a F-sensor on the right side of the cloud is chartered as an example. Further, due to symmetry, it applies to F-sensors on the left side with appropriate position changes. When a child F-sensor node is placed on ar0 it invariably moves on to ar1. It determines upon reaching at ar1, that neither of the sensors in the pair (B1,B0) is in possession of the flag. It moves ahead to ar3 via ar2 and faces a similar situation. Finally it encounters B3 in the pair (B3, B2) that holds the flag. As a consequence, the F sensor moves to C2 and gets a copy of the flag in the process, because it is the first one to do so in that column. The pair (B2, B3) is said to have developed foliage. Further since the pair has let in a F sensor sideways, it also transfers a copy of the flag to B2. That implies the pair (B2, B1) will let in the next immediate F sensor that comes by. Hence as the foliage builds up as depicted in figures 33 and 34, the flag get transferred down the column. In the end B0 gets the flag. Foliage construction is further extended by placing sensors in the third and subsequent columns on either side of the trunk.

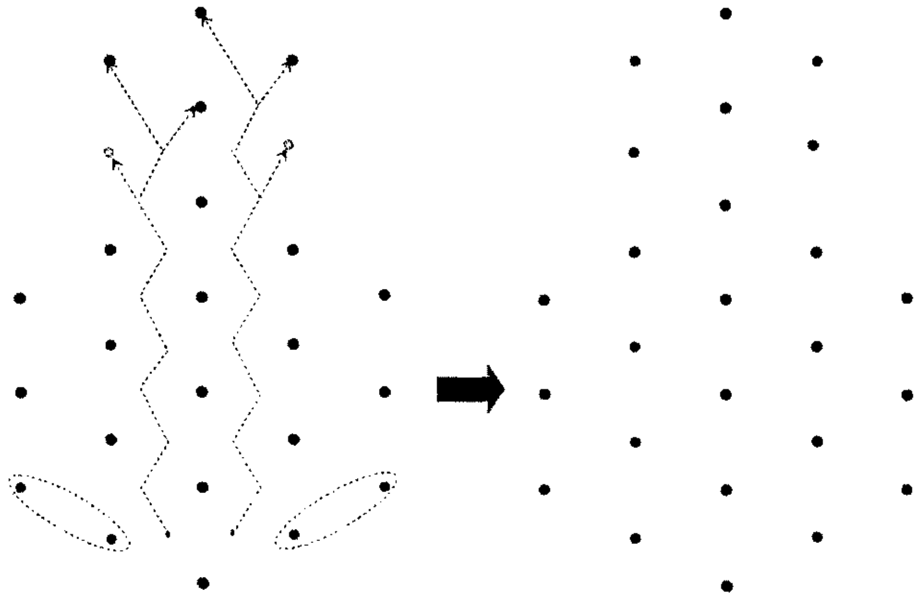


Fig. 35: Flag transfer during cloud extension.

As mentioned earlier, a larger cloud can be built from a fully completed smaller cloud. To enable such a feature, whenever a T-sensor pair lets another T-sensor pass by, it delivers the flag up to the next T-sensor in the column. Therefore during any trunk extending exercise of a completed sensor cloud, the flag ends up at the topmost T-sensor in the column as shown in figure 35. In addition to that, when any flag-possessing axis-located T-sensor relinquishes the flag, all the axis sensors possessing the flag purge it.

Transfer of a flag can occur only under following conditions. Of course, it goes without saying that a flag is needed to pass on one. The prepositioned T-sensors are the only ones to possess the flag to start with.

- From a steady T-sensor to a passing T-sensor going up in the column
- From a steady sensor to its immediate lower sensor in the column if it has let in a child sensor or had let in one earlier.

- From a steady sensor in the topmost pair in the column to a child sensor that it let pass sideways

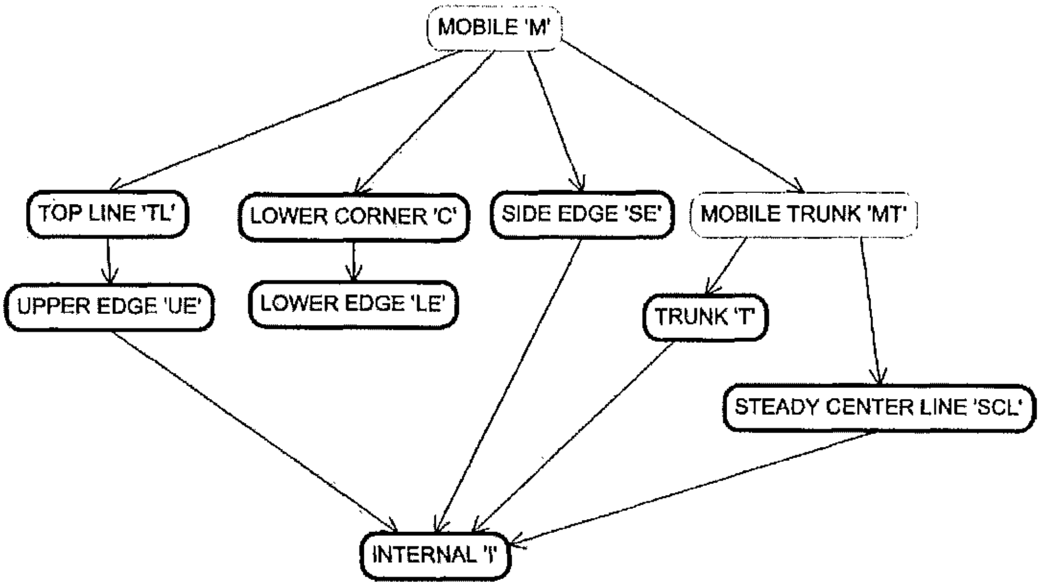
4.1.2 Changing state technique

This technique envisages sensors with varying state information, instead of predetermined static ones. All sensors start out as 'mobile' sensors and are dropped one after another into the primordial cloud. As they wade through the sensors, they take up positions wherever required in such a way that the cloud is formed in a symmetric and organised manner. During this process, they change their state information condition to the destination that they are headed to. Further, whenever a sensor attains a steady state, it invariably leads to change in states of a couple of surrounding sensors too.

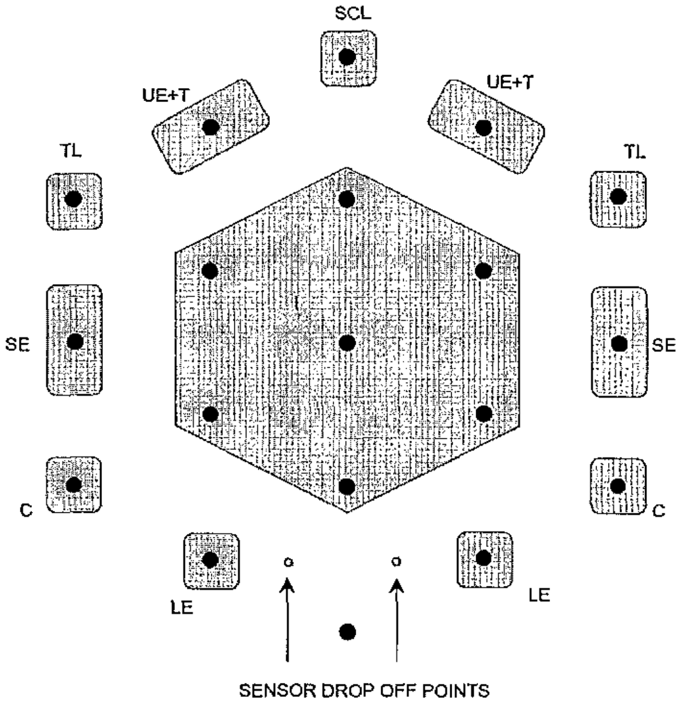
The state flow diagram in figure 36(a) summaries all the states that a mobile sensor can take up subject to the conditions. The states correspond to the locations of the steady sensors. All mobile sensors start out as 'M'. If it is deemed necessary that they should end up on the trunk, they change their state to 'MT'. If they end up positioning themselves on the edges of the cloud, their state pertains to 'LE', 'SE' or 'UE' corresponding to lower, side and upper edges. In addition, 'SCL' refer to those on the central column and 'ST' relates to the trunk columns on either side of 'SCL'. 'C' state refers to corner position on the lower edges of the cloud. Sensors within a cloud are termed as 'I' or Internal ones.

The prepositioned sensors are placed as shown in figure 36(b). Once the sensors are dropped off at their starting points, they begin to move along the edges of the cloud. Right from when they start, the mobile 'M' sensors, correspond to their neighboring steady sensors and act on the responses. The steady sensors respond in such a manner that they send the mobile sensor to the next steady sensors along the edges. So it will appear as if the mobile sensors hop from one point to another, and at each point all they have to have to do is wait for approval to hop on the next point. Refer to figure 37 for paths traversed by mobile sensors to extend the cloud from order 3 to order 4. The messages that transpire depend on the type of sensor that the mobile sensor is in contact with.

Figures 38, 39, 40, 41,and 42 depict communication between a mobile sensor and a lower edge sensor, a corner sensor, a side edge sensor, a trunk sensor, and an upper edge sensor respectively. Figure 43 shows interaction between a mobile



(a) State flow for Changing state technique



(b) Primordial Cloud

Fig. 36: Changing State Technique.

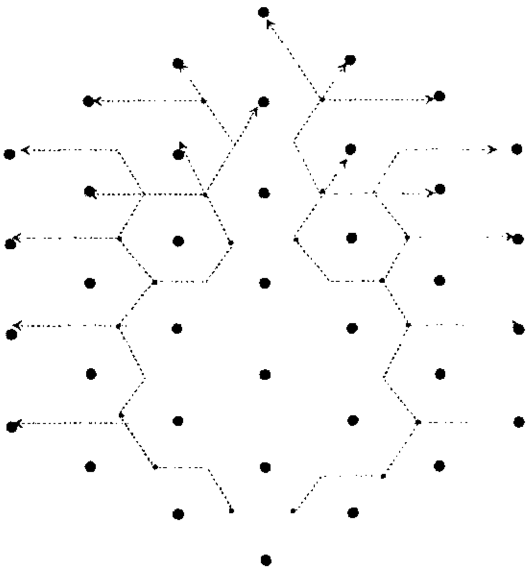


Fig. 37: Cloud Formation.

trunk sensor and a steady trunk sensor.

Both the techniques support dropping in sensors continuously without having to wait for the previously dropped sensor to reach its destination. This feature arises from the fact, that at each of the 'approval seeking points', the mobile sensors are let forward only when they receive the acknowledgement from the next ahead in line mobile sensor.

The operation of each communication is summarized as follows.

- Lower Edge Sensor-Mobile Sensor(Fig 38): This operation is fairly simple and every injected mobile senosor undergoes this transmission first. The act is first initiated by the mobile sensor in the form of a request to a lower edge sensor asking for approval to move ahead. The permission is granted only when the next spot in line is vacated by the preceding mobile sensor.

- Corner Sensor-Mobile Sensor(Fig 39): When a corner sensor makes contact with its first mobile sensor, it marks it and the next two in line as trunk sensors in order to form the trunk. In subsequent transactions, it acts like a lower edge sensor passing mobile sensors one by one ahead in line. As the foliage builds up, there comes a time, when the corner sensor has to relinquish its status and change to a lower edge sensor.

- Side Edge Sensor-Mobile Sensor(Fig 40): Side edge sensors behave like lower corner edge sensors, the only difference being that side edge sensors mutate to internal sensors when the foliage build up overwhelms them.

- Steady Trunk Sensor-Mobile Sensor(Fig 41): During this operation, the foliage first starts building up. When the first columns of foliage is finished, the trunk sensors change their status to upper edge sensors.

- Upper Edge Sensor-Mobile Sensor(Fig 42): Upper edge sensors are the same as the side edge sensors in all respects. They bide away their time passing mobile sensors along the line and end up as internal sensors eventually.

- Steady Trunk Sensor-Mobile Trunk Sensor(Fig 43): This communication transaction is different from all others. It involves positioning the mobile trunk sensors, to form partial trunks on either side of the column. Among trunk sensors, there is a unique TopLine (TL) sensor who has the right to initiate foliage build up. Only the topmost trunk sensor in the trunk is granted the TL status.

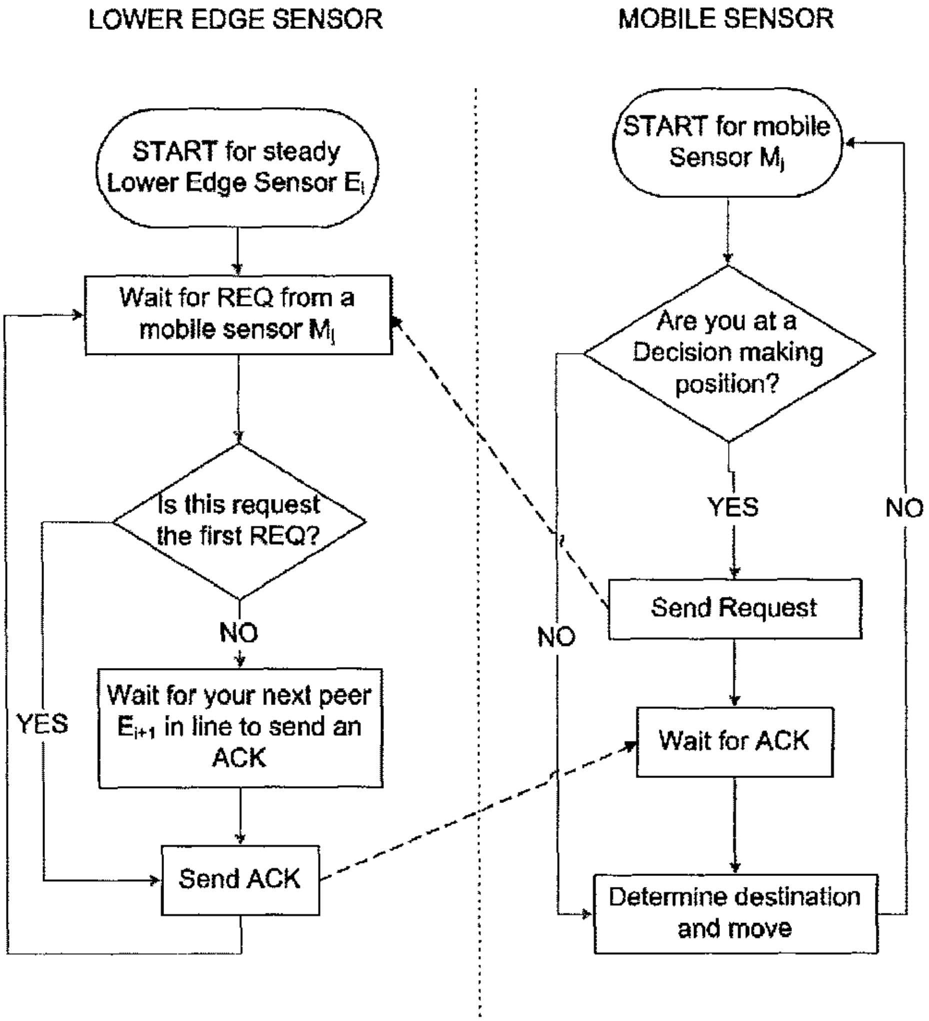


Fig. 38: Lower Edge to Mobile.

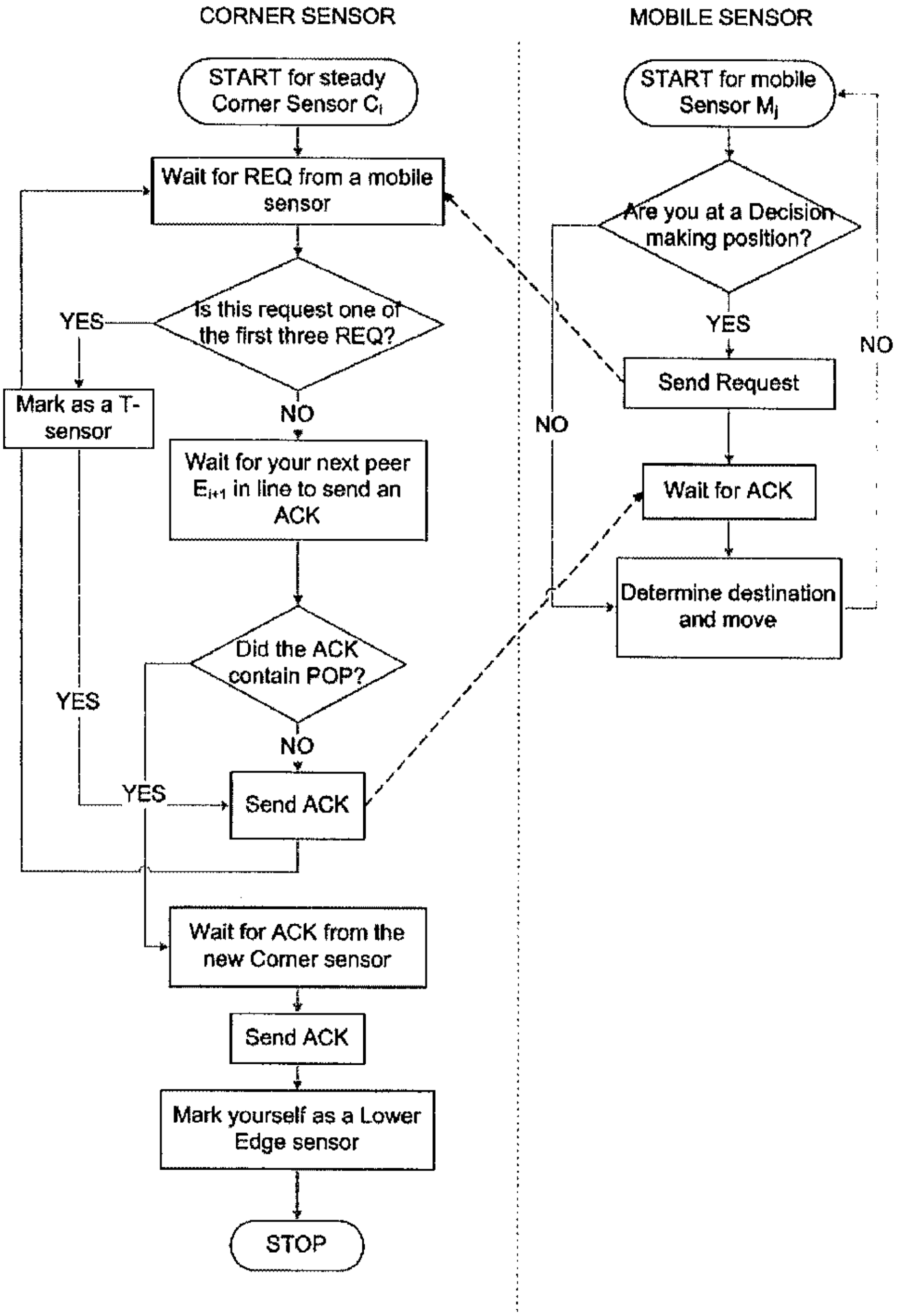


Fig. 39: Corner to Mobile.

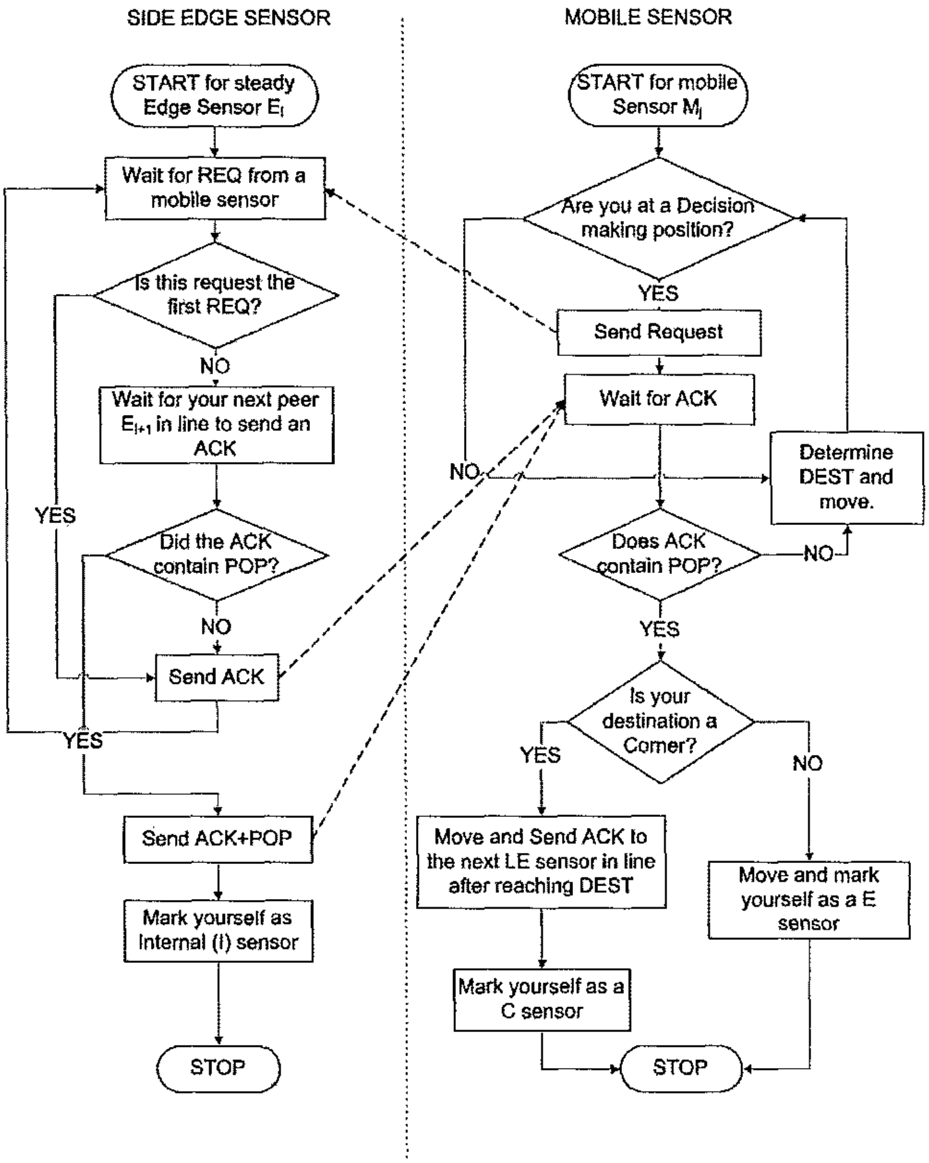


Fig. 40: Side Edge to Mobile.

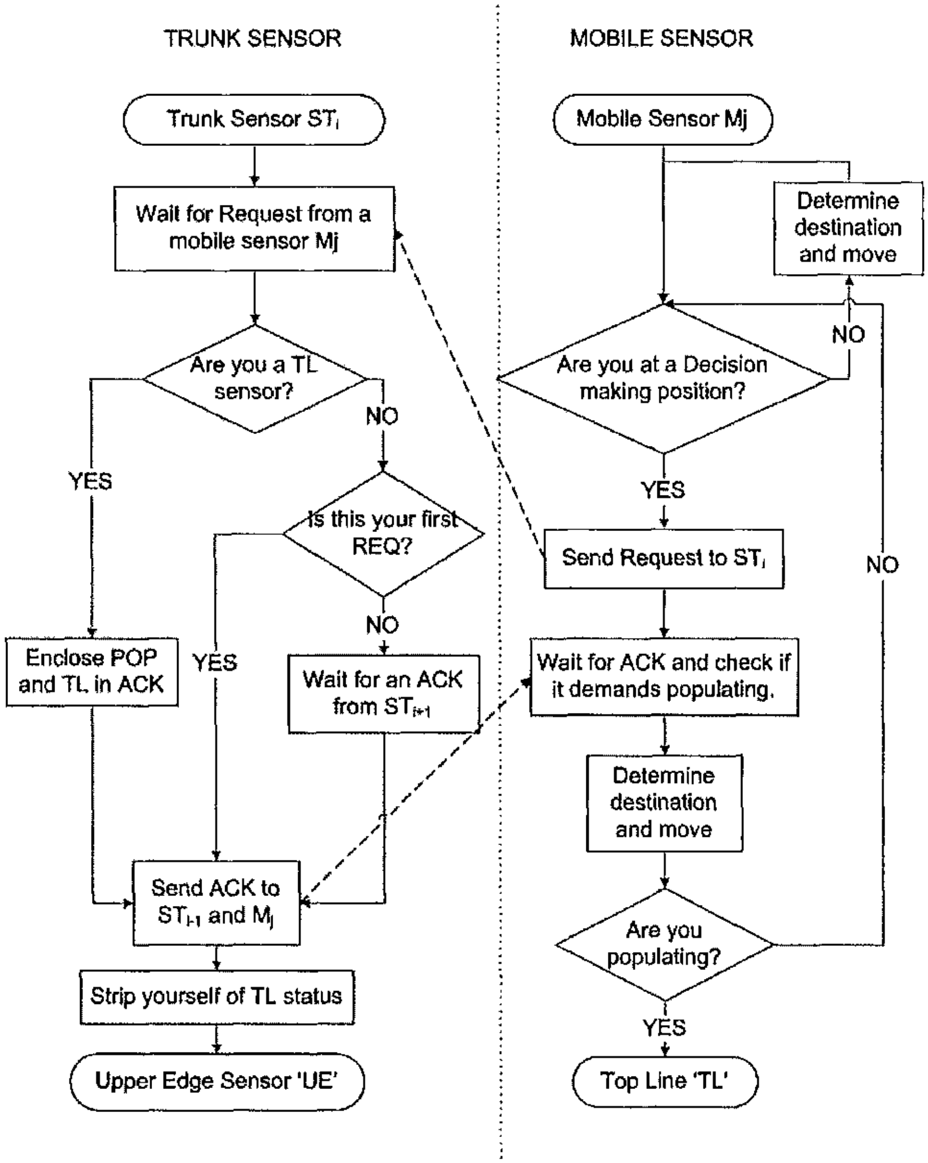


Fig. 41: Steady Trunk to Mobile.

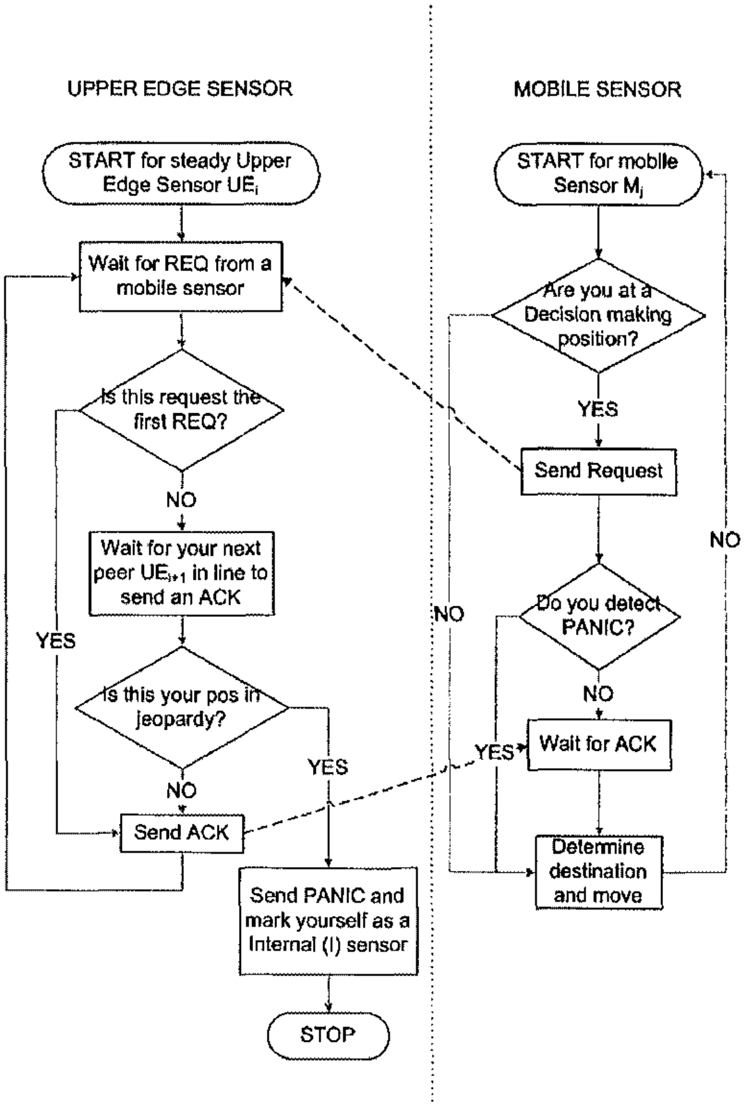


Fig. 42. Upper Edge to Mobile.

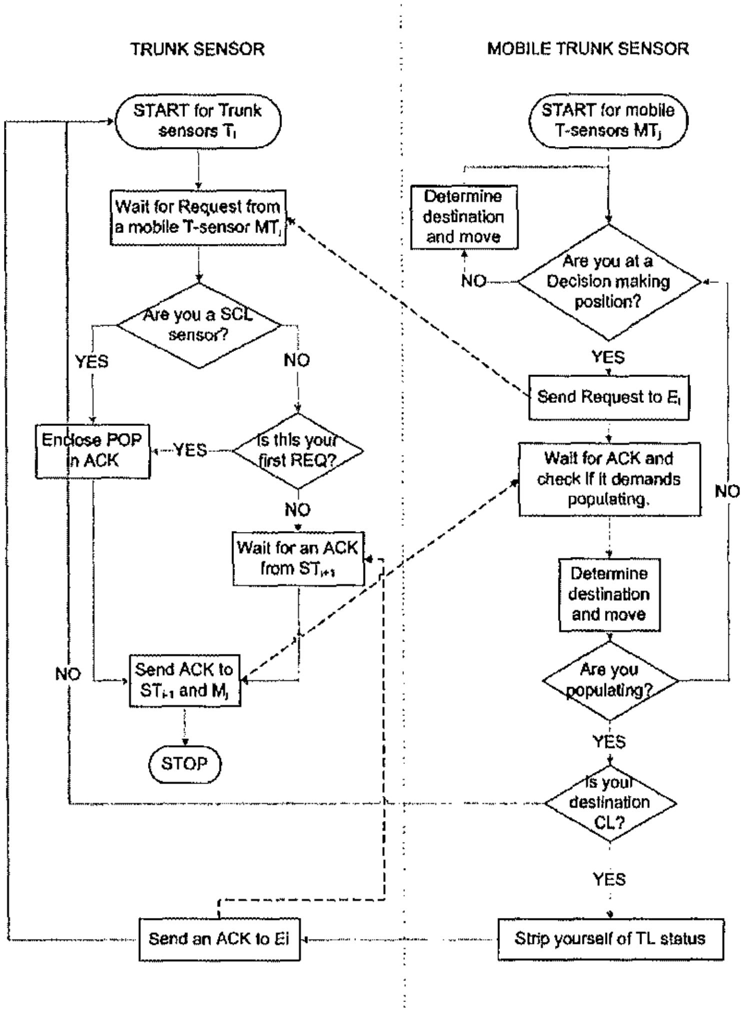


Fig. 43: Steady Trunk to Mobile Trunk.

4.2 CLOUD NAVIGATION

Cloud navigation refers to the process of providing destination coordinates to a cloud, the cloud determining the path and traversing it. For a hexagon cloud, moving it along the axes bisecting the edges or the vertices is easiest. In order to restrict the degrees of motion, the cloud might not take the shortest path to reach the destination. Moving along directions as shown in figure 44 is simple due to symmetry in motion. Each node travels the same unit distance during its turn and each moving sensor has the same angular placement with its reference stationary sensors as shown in figure 45. However, translation along any other axes of motion might result in unequal movements for all sensors and due to the absense of harmony or monotonocity, movement paths become severly complex to compute. To avoid this complexity, a cloud might have to resort to a stair case type of route, with each step of the stair case conforming to that axis of motion closest to the 'line of sight' path.

4.2.1 Sweeping

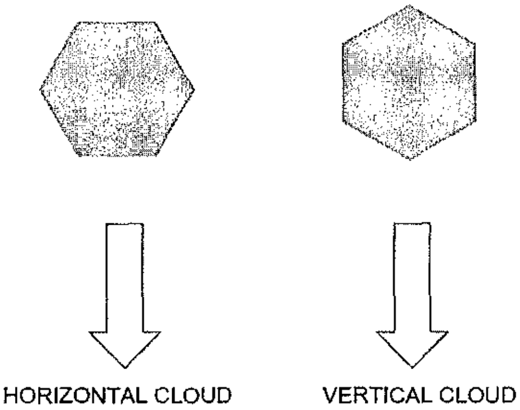
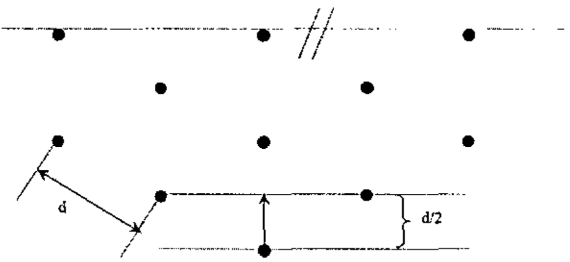
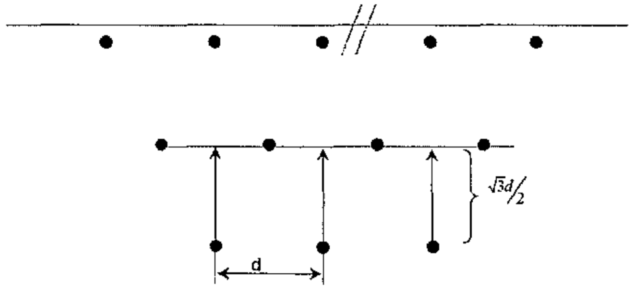


Fig. 44: The two directions of motion.

The hexagaon cloud can travel either way as shown in Figure 44. For simplicity, the two orientations will be referred to as vertical and horizontal clouds. Figure 45 presents the unit distances that the sensors traverse at one time. Figure 49 depicts the various stages involved in translation. To start with, the back edge of the cloud moves forward until it is inline with the second line. The second line then moves ahead to take up places among sensors on the third line. Next,



(a) Vertical Cloud



(b) Horizontal Cloud

Fig. 45: Unit step distances.

the first and third line move ahead to merge with new third and fifth lines. This kind of motion continues until the each rows of the cloud contains two lines of sensors interleaved. The resulting compressed form then traverses across the field and upon reaching the destination, it unravels in the exact opposite manner it contracted itself. One restriction that results in such motion is that at any time, a sensor is only allowed to move relative to two fixed sensors ahead of it on either side of its path. As can be seen in figure 49 there are 3 phases in cloud motion, compress, traverse and decompress.

4.3 RECOVERY

Recovery refers to the process of calling back the cloud to the home base after completion of intended tasks. Usually, it differs from navigation in that the path to the destination is chosen to be the shortest rather than a sweep path. Further, a cloud upon reaching the base may break up into its constituent sensors to be later built up into a cloud of another shape or size, or may reinitiate the scanning process.

4.4 ANALYSIS

4.4.1 Simulation

The flag technique to build clouds has been simulated. The simulation computes the destinations of the sensors and the paths they traverse to reach their destination. Different kinds of sensors are simulated to communicate with their peers and wait for their acknowledgements. The acknowledgements recieved instruct the mobile sensor of its future course of action. Additionally, the simulation supports mobile sensors to be pumped into the cloud in a serial fashion.

The simulation has been developed in VHDL. Reasons for choosing VHDL were its simplicity and ease of obtaining graphical output. Further, a purely behavioral architecture has been designed since architecture style was of a lesser concern than making the entity work. Therefore, with a behavioral architecture, the code could easily be developed to follow along the lines of the flag technique algorithm.

The program structure exhibits the following pattern. To start with, the primordial cloud consisting of 4 sensors goes live. Their ordinates and sensor types are initialized. Next, mobile sensors are turned on and set off from within this primordial cloud, one sensor from each side of the cloud. Each pair of sensors is followed by a sole core sensor that proceeds towards the construction of the core. Each mobile sensor is given its current ordinates and the lead sensor that they need to get in touch with to move ahead. The last stage in the program is the actual movement of the mobile sensors within the cloud. The mobile sensors then pass requests indicating their type to their respective leads. The lead sensors are the steady sensors. Upon obtaining such a request, a lead sensor decides whether to populate the mobile sensor to a corner of a triad or move it ahead to the next triad of sensors. It then sends back an acknowledgement to the mobile sensor following which the mobile sensor moves to the commanded destination. Thus the train of sensors moves ahead.

The building of a cloud of order 4 has been simulated and its results presented here for illustration purposes. The following picture depicts a snapshot of the simulation.

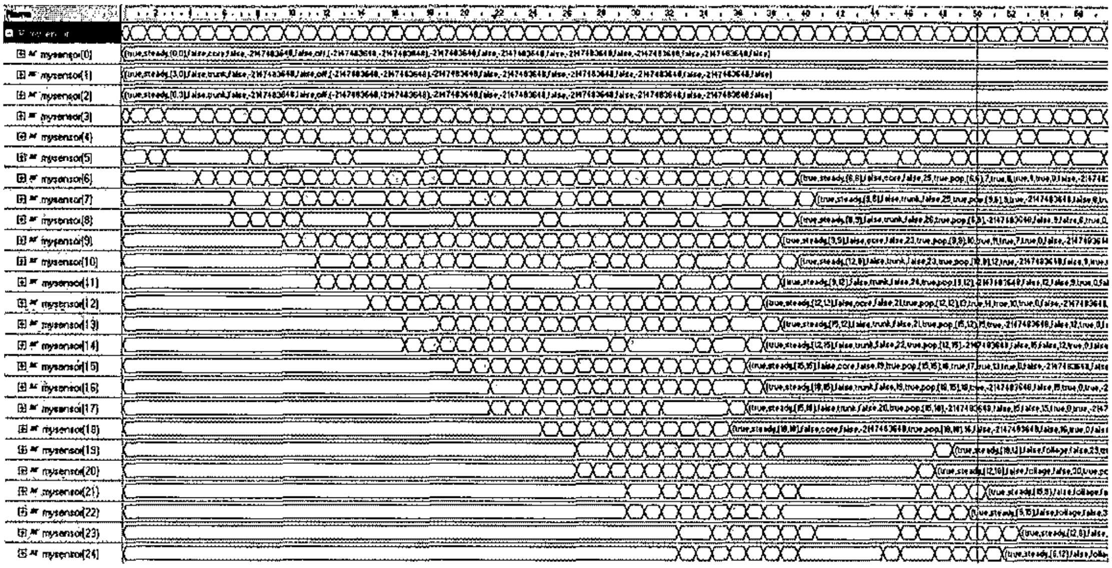


Fig. 46: Snapshot of Simulation.

As seen from the snapshot, sensors are set off sequentially into the cloud. A hexagonal co-ordinates system is adopted for ease and convinience. Each mobile sensor starts off from the self-deployable sensor position as shown in figure 24. At the end of the simulation, the numbered sensors would take up positions as shown in figure 47.

Consider sensor 21 and its behavior during the simulation as depicted in figure 48. Mobile sensors make requests to their lead sensors. These lead sensors then reply back with approvals to proceed and the destination to go to, and only then will the mobile sensors move. There are different attributes to a sensor. The live attribute indicates if the sensor is in use. If the sensor is sent into the cloud, it is said to go live. The status attribute indicates whether the sensors are moving or steady. Similarly, other attributes like ord (ordinates), flag, stype (sensor type), and req (request) are defined for other purposes. The chief attribute is the ordinates one. It indicates the current position of the sensor. All sensors are set off from (2, 1). Hence sensor 21 also starts from (2, 1). As it weaves its path through the cloud, it occupies (4, 2), (5, 4),.and ends up in (15, 9). The mobile state lasts as long as it is not asked to populate. A sensor is required to populate when it recieves the 'pop' command in the acknowledgement. The rleadid attribute is the list of the sensors that it makes requests to during its

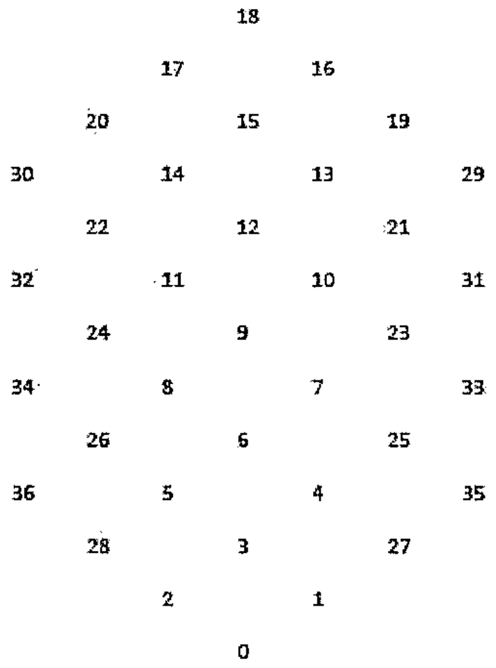


Fig. 47: Result of Simulation.

travel.

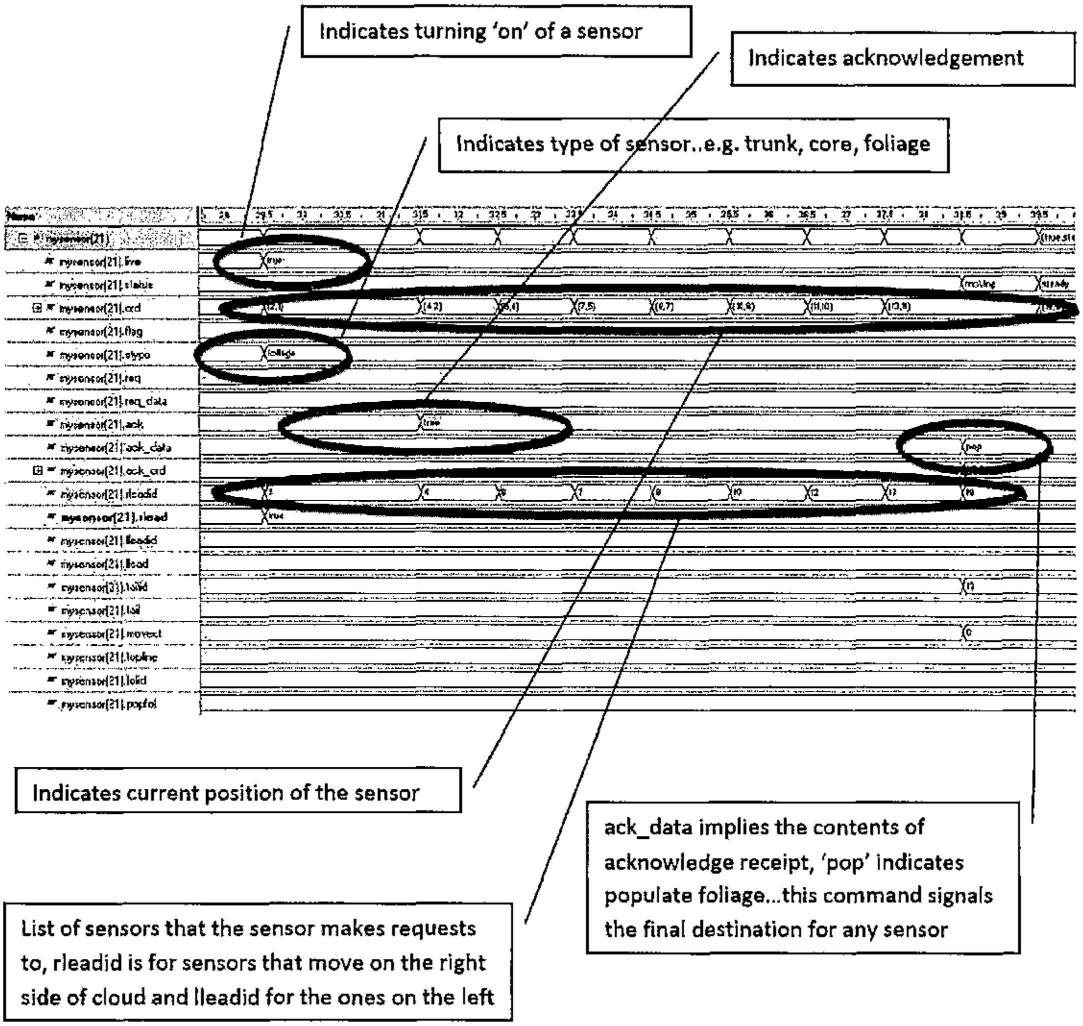
The diagram presented in figure 48 is a typical one. The timelines of all sensors are similar to this one, the differences being only with the sensor type and the path traveled.

The simulation results indicate that the flag technique algorithm is indeed a successful way of building up a hexagonal cloud of sensors. No interruptions or deadlocks surfaced during the simulation. The final co-ordinates of sensors match up with the ones determined by the paper and pencil method. The simulation was run for clouds of different sizes and similar results were obtained.

4.4.2 Translation Time

Refer to figure 49 for the following analysis of translation times of the clouds. When a sensor needs to move, it seeks approval to set off from its forward peers by way of sending requests and recieving acknowlegdements. The time consumed in such communication is done away with in this analysis for simplicity.

Let D_{TOTAL} be the distance required by a cloud to traverse. A cloud executes



Indicates turning 'on' of a sensor

Indicates acknowledgement

Indicates type of sensor..e.g. trunk, core, foliage

Indicates current position of the sensor

ack_data implies the contents of acknowledge receipt, 'pop' indicates populate foliage...this command signals the final destination for any sensor

List of sensors that the sensor makes requests to, rleadid is for sensors that move on the right side of cloud and lleadid for the ones on the left

Fig. 48: Simulation of Sensor Numbered 21.

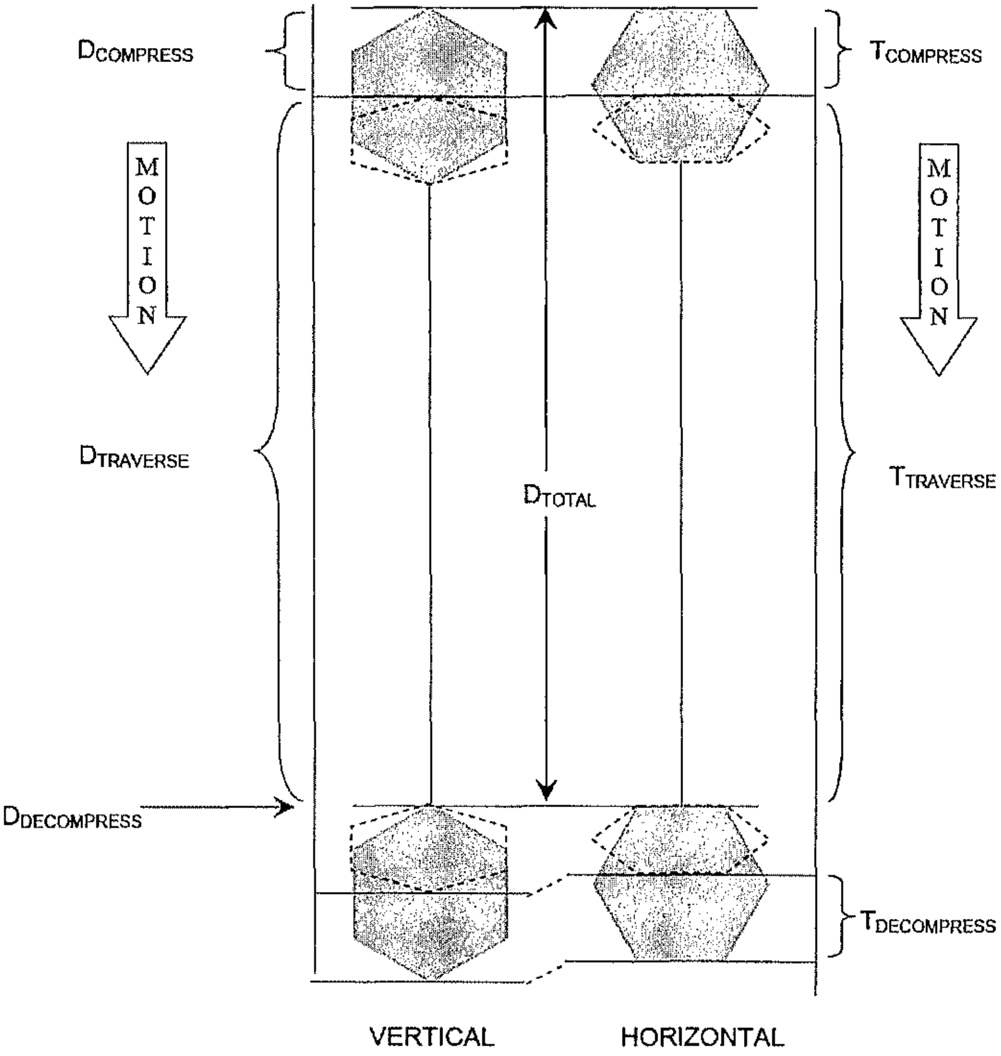


Fig. 49: Analysis of Cloud Translation.

motion in three phases, compression, translation and decompresson. The terms $X_{COMPRESS}$, $X_{TRAVERSE}$, and $X_{DECOMPRESS}$ relate correspondingly to the three phases. Hence, $D_{COMPRESS}$, $D_{TRAVERSE}$, and $D_{DECOMPRESS}$ imply distances travelled by the back edge or corner of the cloud during each of the three phases. And $T_{COMPRESS}$, $T_{TRAVERSE}$, and $T_{DECOMPRESS}$ imply times taken to execute the three phases. Let k be the number of steps that make up D_{TOTAL} and s be the speed of a sensor. As defined in section 4.1 d and n are the inter sensor distances and order of a cloud respectively. The cloud moves in such a fashion that while half of the lines are moving ahead, the other half stays put to act as reference points. Adjacent lines alternate between moving and remaining steady.

Consider the vertical cloud. One sensor stride is $\frac{d}{2}$ units long as shown in figure 45(a). That implies $k = \lfloor \frac{2D_{TOTAL}}{d} \rfloor$. Distances are measured with the back vertex of the cloud as reference as shown in figure 49. During compression, the back half of the cloud pushes itself against the front half such the cloud height is halved. In doing so, the back vertex sensor travels $2(n-1)$ steps. Since, each motion is followed by a steady pause, $T_{COMPRESS} = 4(n-1)\frac{d}{2s}$ and $D_{COMPRESS} = 2(n-1)\frac{d}{2}$. Moving on to traversing, the cloud needs to travel $k - 2(n-1)$ steps to reach its destination. That implies, $T_{TRAVERSE} = (2k - 4n + 4)\frac{d}{2s}$ and $D_{TRAVERSE} = (k - 2n + 2)\frac{d}{2}$. Finally, the decompression time is the same as compression time, the only difference being that when the last vertex sensor reaches its destination, it does not have to pause and act as a reference sensor for its preceding line. Hence, $T_{DECOMPRESS} = [4(n-1) - 1]\frac{d}{2s}$ and $D_{DECOMPRESS} = 0$ since the back edge does not move at all.

$$\begin{aligned} T_{TOTAL} &= T_{COMPRESS} + T_{TRAVERSE} + T_{DECOMPRESS} \\ &= (2k - 4n - 5)\frac{d}{s} \end{aligned} \tag{2}$$

The distance covered in each of the 3 phases is detailed in Eq. (3).

$$\begin{aligned} D_{TOTAL} &= D_{COMPRESS} + D_{TRAVERSE} + D_{DECOMPRESS} \\ &= \frac{kd}{2} \end{aligned} \tag{3}$$

which is the same as $k = \lfloor \frac{2D_{TOTAL}}{d} \rfloor$, the equation we stated in the beginning.

Consider the horizontal cloud. One sensor stride is $\frac{\sqrt{3}d}{2}$ units long as shown in figure 45(b). That implies $k = \lfloor \frac{2D_{TOTAL}}{\sqrt{3}d} \rfloor$. Distances are measured with the back edge of the cloud as reference. During compression, the back half of the cloud pushes itself against the front half such the cloud height is halved. In doing so, the back edge sensor travels $(n-1)$ steps. Since, each motion is followed by a steady pause, $T_{COMPRESS} = 2(n-1)\frac{\sqrt{3}d}{2s}$. And $D_{COMPRESS} = (n-1)\frac{\sqrt{3}d}{2}$. While traversing, the cloud has to manage $k-(n-1)$ steps. Hence, $T_{TRAVERSE} = (2k-2n+2)\frac{\sqrt{3}d}{2s}$ and $D_{TRAVERSE} = (k-n+1)\frac{\sqrt{3}d}{2}$. Finally, the decompression time is the same as compression time, the only difference being that when the last line reaches its destination, it does not have to pause and act as a reference line for its preceding line. Hence, $T_{DECOMPRESS} = [2(n-1)-1]\frac{\sqrt{3}d}{2s}$ and $D_{DECOMPRESS} = 0$ since the back edge does not move at all.

$$\begin{aligned} T_{TOTAL} &= T_{COMPRESS} + T_{TRAVERSE} + T_{DECOMPRESS} \\ &= (2k+2n-3)\frac{\sqrt{3}d}{2s} \end{aligned} \tag{4}$$

The distance covered in each of the 3 phases in detailed in Eq. (5).

$$\begin{aligned} D_{TOTAL} &= D_{COMPRESS} + D_{TRAVERSE} + D_{DECOMPRESS} \\ &= \frac{\sqrt{3}kd}{2} \end{aligned} \tag{5}$$

which is the same as $k = \lfloor \frac{2D_{TOTAL}}{\sqrt{3}d} \rfloor$, the equation we stated in the beginning.

4.4.3 Deployment Time

Consider the time involved in building up a cloud.

As stated earlier in 4.1.2 on page 46, both the techniques feature dropping sensors continuously into the cloud as soon as the previous one makes way for the next sensor. Hence the total time consumed is the sum of the time to drop all the sensors and the time taken up by the last sensor to reach its destination. Observing figure 50(a) each sensor takes a unit time length $t = \frac{d}{\sqrt{3}s}$ to go from one triad to another. In the vertical cloud, shown in figure 51(a), if the changing state technique is adopted to build the cloud, the time intervals to move from one triad to another can be stated as one of the following series,

- $[2t, 2t, 2t, ...2t], t, [2t, ...2t], t, [2t, ...2t], 3t, [2t, ...2t]$

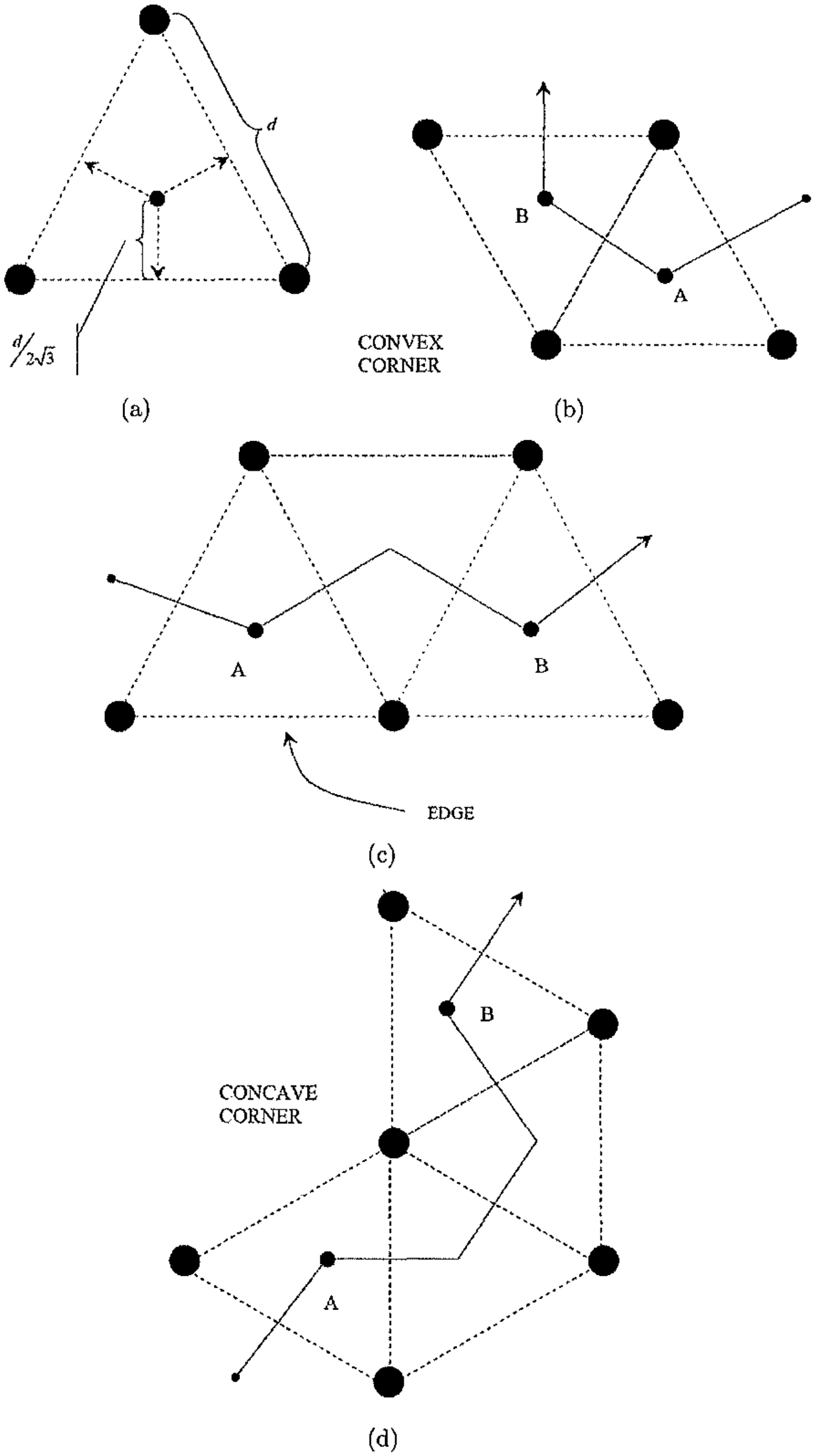


Fig. 50: Analysis of Cloud build up time.

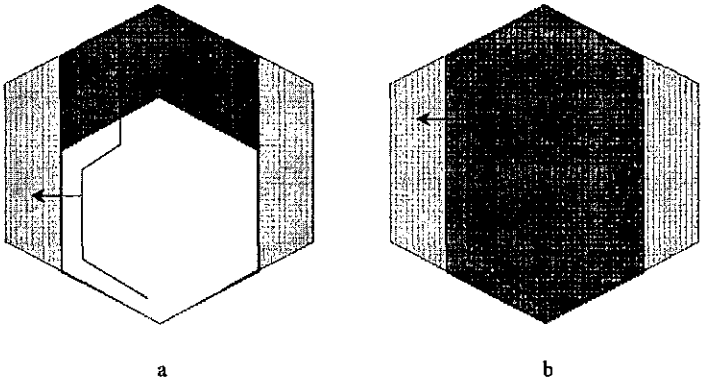


Fig. 51: Deployment Times.

- $[2t, 2t, 2t, ...2t], t, [2t, ...2t]$

The first series correspond to sensors filling up the dark gray area and the second one relate to that in light gray area. In the series, the t's explain the convex corners shown in figure 50(b) and the $3t$ explain the concave corner near the top at the trunk figure 50(d). The rest $2t$ are for the edges shown in figure 50(c). The back log created by the $3t$ is negated by the preceding t. However, a following t cannot compensate for a preceding $2t$. Finally, the drop intervals can be set to $2t$.

If the flag techinque is followed as shown in figure 51(b), the same time interval series would appear as $2t, 2t, 2t, ...2t, t, 2t, ...2t, 2t, 2t$. The lone t is attributed towards the lower convex corner. Hence the drop interval again can be set to $2t$.

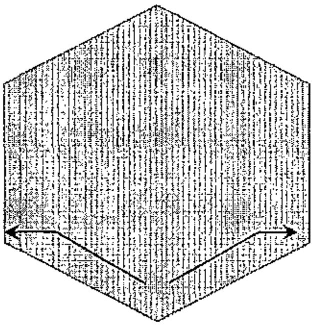


Fig. 52: Drop Times.

It takes $3n^2 - 3n$ sensors to construct a n order cloud. Since sensors can be dropped at regular intervals of $2t$, the total time for cloud build up is the sum of the times taken to drop all sensors and the time taken by the last sensor to reach its destination. Time required to drop all sensors, $T_{DROP} = t(3n^2 - 3n)$ since half the sensors are dropped on either side of the cloud. Further as shown in figure 52, the route of the last sensor is the same in both the techniques. It can be derived by induction that the time taken by the last sensor to reach its final position is $(2n - 3)t$. Hence, total time to build a cloud, T_{CLOUD} can be stated as following.

$$\begin{aligned} T_{CLOUD} &= T_{DROP} + T_{LASTSENSOR} \\ &\approx \frac{d}{\sqrt{3}s}(3n^2 - 3n) + (2n - 3)\frac{d}{\sqrt{3}s} \\ &\approx (3n^2 - n - 3)\frac{d}{\sqrt{3}s} \end{aligned} \tag{6}$$

Hence it can be concluded that the drop times for both the techniques are the same. In the changing state technique, the sensors assume their roles depending upon their position and hence all sensors in the cloud are essentially the same. However, the flag technique reveals simpler routing in exchange for rigid classification of sensors into trunk and foliage sensors. Also observe that the last sensor in both the methods follows the same path and takes the same amount of time to reach its position.

CHAPTER 5

CONCLUSIONS

This thesis presented a sensor cloud, a model of completely connected network of sensors, and observed how it performed tasks. To start with, the different types of sensor clusters developed in the past were discussed. A model of the sensor cloud was then constructed encompassing the characteristics and management of such a cloud. Further, to illustrate the model in greater detail, a plan for a hexagon shaped cloud was drawn out with the different management techniques related to it. Two algorithms for deployment were presented and compared. The times for cloud build up and navigation were computed. The results presented in section 4.4 revealed that the two deployment techniques exhibited the same cloud build up times, even though the ways they did it were markedly different. However, the navigation times varied implying that the cloud oreintation makes a difference.

5.1 CONTRIBUTIONS TO THE FIELD

As mentioned in 1.4, this thesis has put forward the complete model in 3.1 and 3.2. Tasks such as deployment, navigation and recovery are elaborated on. This clear model of a cloud should help the reader visualize the past published ideas of sensor clusters. The specific cloud shape presented in Chapter 4 is shown to follow the model and demonstrate the features. The detailed deployment procedure for the hexagon shape adds to the understanding. The proposed model can be used as an aid to understanding current sensor systems. This perspective helps in critically reviewing related systems. Further, future systems can be developed conforming to the model with individual sections based on past systems.

5.2 FUTURE WORK

Further analysis can be carried out, on determining other ways of deployment that take shorter times for execution. Scalability is another major issue. With the suggested algorithms for deployment, build up times become uncomfortably large for massive cloud sizes. Different cloud shapes and sensor arrangement within a cloud need to be looked into for alleviating this problem. The deployment solution proposed in this paper can be improved by considering dropping in a variable number of sensors at each step of cloud formation. Also, the use of multiple paths would reduce the step delay from the pipeline effect.

Strong assumptions are made on the capabilities of the nodes, in particular the ability of each node to measure the exact range and bearing with respect to its near neighbors. The algorithm can be extended to understand the implications of approximate estimates of range and bearing readings.

Other future enhancements are listed as follows. The clouds suggested are ideal for blanket coverage. However, in order to provide blanket or sweep coverage, the cloud should have a linear orientation. Further, the network should be able to reconfigure itself to form other shapes to avoid or navigate around obstacles. It would also be interesting to devise a way to keep the cloud connected even when some of the nodes cease to function (e.g. due to energy depletion) or are removed (e.g. due to malicious intervention). A simple solution could be that when a node has less than the usual number of neighbors, it moves towards its closest neighbor till it gets connected to some of the neighbor's neighbors.

Error analysis is another attractive add-on. Whenever a cloud locates a region of interest, the cloud takes note of the position by recording the local co-ordinates of the sensor that detected the highest concentration of activity, and moves on. Since a cloud is a collection of sensors and not a monolithic structure, distances traversed by each sensor may vary by small amounts due to differences in trilateration computations. The trilateration computation comes into picture during movement when a mobile sensor moves ahead with reference to two steady ones. The resulting error in translation is a result of the positional errors of the reference sensors and the error in trilateration to determine the mobile sensor's position. Over time, these errors may add up and when the global ordinates for the regions of interest are being computed, these errors must be taken into account.

BIBLIOGRAPHY

[1] Pottie, G. J. and Kaiser, W. J. 2000, "Wireless integrated network sensors". *Commun. ACM* 43, pp. 51-58, May 2000.

[2] V. A. K. Sohrabi, J. Gao and G. J. Pottie, "Protocols for self-organizaion of a wireless sensor network," *IEEE Personal Communication*, pp.16-27, October 2000.

[3] P. R. T. Clouqueur, V. Phipatanasuphorn and K. K. Saluja, "Sensor deployment strategy for target detection," WSNA, 2002.

[4] Y. Zou and K. Chakrabarty, "Uncertainty-aware sensor deployment algorithms for surveillance applications," *Proceedings of IEEE Global Communications Conference* (GLOBECOM), San Francisco, CA, December 2003.

[5] T. L. P. G. Wang, G. Cao, "Movement assisted sensor deployment," Technical Report CS-03-008, Dept of Computer Science and Engineering, Pennsylvania State University, 2003.

[6] G. Wang, G. Cao and T. La Porta, "A Bidding Protocol for Deploying Mobile Sensors," *The 11th IEEE International Conference on Network Protocols (ICNP),* November 2003.

[7] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometrical data structure," in *ACM Computing Surveys* 23, 1991.

[8] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. "Habitat monitoring: Application driver for wireless communications technology". In *Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications* in Latin America and the Caribbean, April 2001.

[9] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. 2002. "Wireless sensor networks for habitat monitoring". In *Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications* (Atlanta, Georgia, USA, September 28 - 28, 2002). WSNA '02.

[10] Liu, T., Sadler, C. M., Zhang, P., and Martonosi, M. 2004. "Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet" In *Proceedings of the 2nd international Conference on Mobile Systems, Applications, and Services* (Boston, MA, USA, June 06 - 09, 2004). MobiSys '04.

[11] Schwiebert, L., Gupta, S. K., and Weinmann, J., "Research challenges in wireless networks of biomedical sensors". In *Proceedings of the 7th Annual international Conference on Mobile Computing and Networking* (Rome, Italy). MobiCom 2001.

[12] S. de Vlaam, "Object tracking in a multi sensor network," Delft University of Technology, Computer Engineering, Parallel and Distributed Systems, August 2004.

[13] D. Gage, "Many-robot mine countermeasure search systems," *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium*, Monterey CA, April 1995.

[14] D. W. Gage, Ed., "Command Control for Many-Robot Systems", vol. 10. *Proceedings of AUVS*-92, Huntsville AL, June 1992.

[15] A. M. Flynn, "Gnat robots (and how they will change robotics)," *Proceedings of the IEEE Micro Robots and Teleoperators Workshop*, Hyannis MA, November 1987.

[16] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", *IEEE Proceedings of the Hawaii International Conference on System Sciences*, pp. 1–10, January 2000.

[17] D. W. Gage: "Sensor Abstractions of Support Many Robot Systems", *Proc. of SPIE Mobile Robots VII Conference*, Vol. 1831, pp.235-246, 1992.

[18] A. Howard, M. Mataric, and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, June 25-27 2002.

[19] Sameera Poduri and Gaurav Sukhatme, "Constrained coverage for mobile sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr. 2004.

[20] N. Heo and P.K. Varshney, "A Distributed Self Spreading Algorithm for Mobile Wireless Sensor Networks," *IEEE Wireless Comm. and Networking*, pp. 1597-1602, Mar. 2003.

[21] Y. Zou and K. Chakrabany, "Sensor deployment and target localization based on virmal forces," *INFOCOM, 2003.*

[22] R. Want et al. "The Active Badge Location System". *ACM Transactions on Information Systems,* January 1992.

[23] P. G. Estrin D, Girod L and S. M, "Instrumenting the world with wireless sensor networks," in *ICASSP IEEE INT CONF ACOUST SPEECH SIGNAL PROCESS PROC.* Vol. 4, 2001.

[24] L. Prasad *et al.*, "Functional characterization of sensor integration in distributed sensor networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 1082–1087, Sept./Oct. 1991.

[25] S. Krishnamachari, B Iyengar, "Distributed bayesian algorithms for fault tolerant event region detection in wireless sensor networks," *Computers, IEEE Transactions on,* vol. 53, no. 3, pp. 55-78, March 2004.

[26] B. Krishnamachari and S. Iyengar, "Self-organized fault-tolerant feature extraction in distributed wireless sensor networks," Autonomous Networks Research Group (ANRG) Technical Report 02-003, September 2002.

[27] T. L. P. Guiling Wang, Guohong Cao and W. Zhang, "Sensor relocation in mobile sensor networks," *IEEE INFOCOM,* March 2005.