A DYNAMIC DATA GATHERING IN VIDEO BASED WIRELESS SENSOR NETWORKS

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul Ştiinţa Calculatoarelor
de către

ing. Codruţa Mihaela Istin

Conducător ştiinţific: prof.univ.dr.ing Horia Ciocarlie

Referenţi ştiinşifici: prof. Dr. Ing. Victor Valeriu Patriciu
prof. Dr. Ing. Dan Dumitru Burdescu
prof. Dr. Ing. Ştefan Holban
Alex Doboli, Phd, Associate Professor

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul Departamentului de Calculatoare al Universităţii „Politehnica” din Timişoara.

Mulţumiri deosebite se cuvin conducătorului de doctorat prof.dr.ing. Horia Ciocarlie care m-a indrumat inca de la inceputul studentiei si mi-a condus pasii formandu-ma atat profesional dar si uman. Domnul prof. dr. ing. Horia Ciocarlie a fost cel care cu incredere m-a sustinut de-a lungul perioadei doctorale atat financiar pentru a putea sa aprticip la un numar considerabil de conferinte cat si stiintific si moral. Ii multumesc pe aceasta cale pentru tot.

De asemenea ii multumesc si dl. conf. dr. ing. Dan Pescaru, care m-a primit cu caldura in grupul de cercetare pe care il coordoneaza. Am invatat de la dumnealui ce inseamna dedicatia pentru cercetare, pasiunea in munca si mai presus de toate ii multumesc pentru incurajarile si modalitatea prin care a reusit sa ma motiveze pentru a putea scrie si publica la manifestari stiintifice valoroase.

In mod special doresc sa ii multumesc lui Alex Doboli, Assoc. Prof la State University of New York, Stony Brook, pentru increderea si sustinerea pe care mi-a oferit-o neconditionat dar si pentru indrumarile si preocuparea constanta care a avut-o pentru formarea mea ca cercetator. Cu ajutorul dumnealui am reusit sa scriu si sa public la reviste de prestigiu. Tot acest ajutor m-a format pentru o cariera atat stiintifica cat si dicactica.

Nu in ultimul rand le multumesc familiei, prietenilor apropiati si foarte apropiati, cunostintelor si lui Dumnezeu care m-au ajutat, sustinut si ghidat pentru a devei mai buna si pentru a duce la bun sfarsit aceasta etapa importanta din viata si cariera mea.

Istin, Codruta Mihaela

Titlul tezei

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 2011, 149 pagini, 72 figuri, 10 tabele.

Cuvinte cheie: video wireless sensor networks, FoV recovery, dynamic data gathering, redundany, prediction, events, surveillence

Rezumat,

The theme of this thesis regards the domain of video based wireless sensors and proposes two algorithms fost real time field of view recovery with minimum of resources used. This implies maintaining the coverage at a high level while preserving and prolonging the lifetime of the network.

The algorithms are performant and make use of novative concepts such as the one of redundancy, redundancy groups, events and prediction. These algorithms can be used to analyze both micro and macro traffic situations. Furthermore, in order to validate these algorithms, Monte Carlo simulation was performed and a simulator **Simulo** was implemented.

The simulator's unique characteristics represented the motivation for implementing a new simulator. Among these characteristics to be mentioned are the fact that Simulo is capable of simulating both micro and macro traffic situations, is capable of saving the previos test cases so that the same data sets to be used for other algorithms in order to obtain an objective comparison. The simulator was implemented considering a mathematical model that was developed for this purpose.

All in all, the thesis proved the value of the proposed algorithms.

Table of Contents

List of Figures

List of Tabes

List of Acronims

FoV	Field of View
MC	Monte Carlo
WSN	Wireless Sensor Networks

Abstract

Wireless sensor networks have become essential in our daily lives. However, the domain is still young and challenges wait to be answered with suitable solutions. This thesis has several goals. The first objective aims to provide a thorough theoretical background regarding quality coverage in wireless sensor networks as well as the metrics used to evaluate the coverage quality. Frameworks used for creating different testing scenarios are described together with their limitations. More and more, the area of wireless sensor networks is integrated in the vast domain of cyber physical systems. This implies a strong link between the cyber world and the physical one. The cyber physical systems field is in its early age, but it promises to incorporate a lot of domains. From the wireless video sensor networks perspective, there still is a gap between the cyber and the physical world. Data gathering and its communication has to be done in real time. This fact includes the domain of algorithms for video based wireless sensor networks in the area of Kinetic data structures. The limitations come from the limited energy in contrast with the huge amount of data to be processed and transmitted to the sink, from the fact that the data accuracy is debatable not to mention the communication shortage. The next step was to investigate these faults and their solutions as they appear in literature. Furthermore I addressed the problem of coverage quality by proposing new metrics for video based wireless sensor networks. This represents the second goal of this thesis.

Further more, solutions to the addressed issues are proposed. In this direction, the first goal is to present the theoretical foundation on the one hand and the practical implementation corresponding to the theoretical part, on the other hand. Another important aspect also present in this report is the validation of the proposed algorithms. The performance of the proposed solutions was realized by Monte Carlo Simulation, and by comparing with theoretical solutions and also the comparison between algorithms themselves.

In addition to this, a simulator called Simulo was realized. Simulo is capable of simulating different traffic situations with desired car types, monitoring road traffic, setting driving behavior and setting the types of sensors by their shape. The value of the presented algorithms is validated bysimulation with Simulo and by a considerable number of articles published at conferences and transaction.

1. Introduction

1.1. Standpoint on Wireless Sensor Networks

Wireless sensor networks have become indispensable in a variety of areas. There are different types of wireless sensors, but all have some limitations. One of the problems that are still in the top of research in this domain concerns the energy consumption of the nodes. This implies the longevity of the networks' lifetime. In a wireless sensor network, not all nodes die simultaneously. It is rather difficult to predict witch node is going to die and at what time. Furthermore, when a node becomes unuseful, coverage quality drops. Analyzing the influence of the sensors' behavior involves the development of different metrics. Therefore, coverage metrics have become another issue that is intensely studied. Metrics are needed to evaluate the performance of the network and make the suitable choices to keep that performance at suitable levels. In order to obtain an efficient coverage, redundancy is needed, but redundancy also means more cost and more communication that also translates into greater energy consumption.

A particular case of sensors is video based sensors. This type of sensors differs in many aspects from the usual sensors. They need local data processing due to the huge amount of data that is captured and not all of it is of interest and needs to be transmitted. There are important features that differs video based sensors from the other ones. The great majority of video-based sensors are unidirectional. Due to this aspect, almost no metric used in the case of regular sensors is applicable in the case of video-based sensors. The applications that need video-based sensors are various and many necessitate accurate data that can be provided in most of the cases by using a larger degree of redundancy. The area covered by a video sensor is smaller than the one obtained by a regular sensor and the number of sensors to efficiently cover an area is high. Also, when a sensor dies, the performance in terms of coverage is affected significantly, if the redundancy between sensors is low. The cost of a wireless sensor network is estimated taking into consideration the number of nodes and their lifetime that is strongly connected with the amount of transmitted information and with the functioning period. The performance of the wireless sensors is estimated in terms of the amount of data collected and transmitted, data accuracy, real time data acquisition and network's lifetime.

If the sensor network is used for target tracking, the accuracy of the collected data may be vital. This implies that both the deployment of the sensors as well as their connectivity and the lifetime of the network are important. In a wireless sensor network, not all nodes die simultaneously. It is rather difficult to predict witch node is going to die and at what time. Furthermore, when a node becomes unuseful, coverage quality drops. The lifetime of a network is also influenced by the type of sensors that are used for performing their task. If they are regular sensors, they might have a longer lifetime than a video sensor due to the fact that in the case of video sensors, the tiny video is an additional device that needs energy in order to be able to collect data and send it to the sink. The proposed algorithms aim to research the topics mentioned above. The lifetime of the network is computed with Simulo and the results show that is also prolonged when the proposed algorithms are used. The tests, performed by simulation are to be presented in detail in the next chapters. One of the important factors that make our approach different is the fact that we perform all these analysis in the presence of dynamic obstacles. It will be detailed the fact that the dynamism can be modeled by setting the drivers' behavior. This will influence the patterns of dynamic obstacles and brings the simulation a lot closer to reality.

Even though the domains where the sensors are used differ, the basic sensor structure remains the same. Figure 1.1. shows the main components of a sensor. It is important to know the role of each component in order to use the sensors' capability in an optimal way. This is why I will shortly present each component.

- Sensing Unit

Role: to sense the specific factors from the areas where the sensor was deployed

Components: Sensor – generates an analog signal that corresponds to the sensed factor

ADC - converts the analog signal received from the sensor into a digital one

- Processing Unit

Role: to interconnect with other sensors

Components: Processor – processes the data in order to transmit only the relevant information to the sink

Storage – keeps the data in order to be processed by the local Processor

- Transceiver

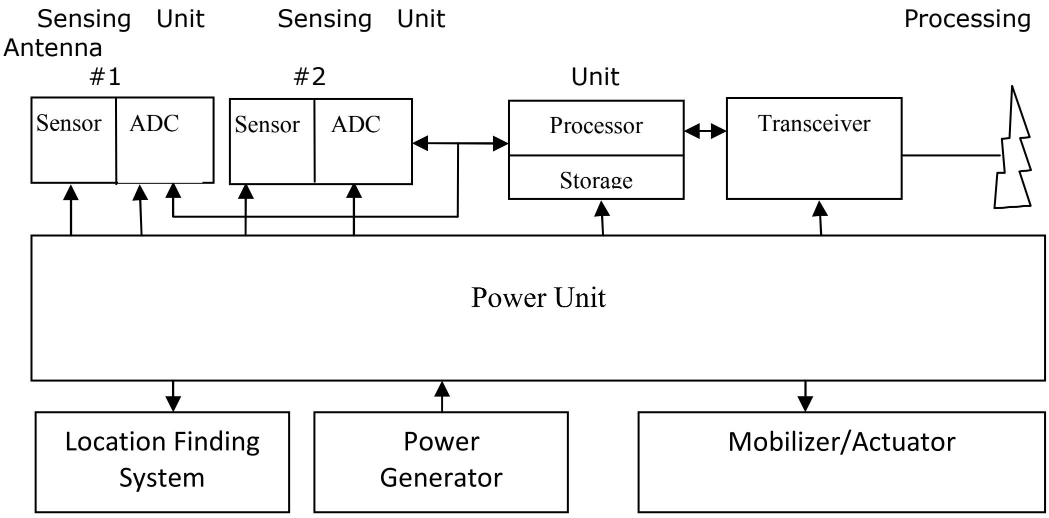
Role: transmits the processed data to the sink by means of radio, laser or Infrared

- Power Unit

Role: provides power to the devices by the use of batteries that can be rechargeable or non-rechargeable

In the case of video based sensors, from the architectural point of view the sensing unit is represented by a small video camera.

A wireless sensor network is made out of a large number of sensors that are potentially capable to locate themselves and create the topology in case of random deployment, to communicate with each other and to communicate with the sink, to collect and transmit data on the optimal path between themselves and between themselves and the central unit using specific routing algorithms, to share communication channels, to be able to decide which data is relevant and perform local data processing, etc.



1.1. Sensor components. Acknowledgement: adapted from [1]

1.2. Applications Using Video Based Wireless Sensor Networks

Wireless Sensor Networks (WSN) has gained a lot of attention due to their efficiency in getting data from observing external factors. Sensors are of different types and used in different domains and applications. They are very large collections of tiny sensor nodes that form ad hoc distributed sensing and data propagation networks that collect detailed information about the physical environment [2]. The applicability of WSN is widespread from military applications to health care and different types of monitoring. In a usual scenario, these networks are largely deployed in areas of interest (such as inaccessible terrains or disaster places) for fine-grained monitoring in various classes of applications. Various classes of applications with WSN are presented in [2]. The reason for the increasing usage of WSN is their low cost in comparison with the work they do. WSN has proved their efficiency by the way they collect data even from hard inaccessible fields such as deep into waters or high mountains or even in spying actions, where placement is vital. The most common type of sensors is a device that measures or detects a real-world condition, such as motion, heat or light and converts the condition into an analog or digital representation [1].

The constantly increasing interest and usage of sensor networks in many diverse domains proved their importance and utility. This aspect determined the researchers to develop, overcome and improve the existing technology in this domain. A recent study made by the authors in [3] presents the most analyzed aspects. Their classification took into consideration the frequency and the number of published papers in every research topics: deployment 9.70%, target tracking

7.27%, localization 6.06%, data gathering 6.06%, routing and aggregation 5.76%, security 5.76%, MAC protocols 4.85%, querying and databases 4.24%, time synchronization 3.64%, applications 3.33%, robust routing 3.33%, lifetime optimization 3.33%, hardware 2.73%, transport layer 2.73%, distributed algorithms 2.73%, resource-aware routing 2.42%, storage 2.42%, Middleware and task allocation 2.42%, calibration 2.12%, wireless radio and link characteristics 2.12%, network monitoring 2.12%, geographic routing 1.82%, compression 1.82%, taxonomy 1.52%, capacity 1.52%, link-layer techniques 1.21%, topology control 1.21%, mobile nodes 1.21%, detection and estimation 1.21%, diffuse phenomena 0.91%, programming 0.91%, power control 0.61%, software 0.61%, autonomic routing 0.30%. These statistics show the fact that on the top of the hierarchy is placed the deployment of the sensors and on the second place is target tracking. Target tracking implies video based sensors. The research in this specific area are so intense because the optimum balance between real time requirements, processing time and accuracy of data has not yet been found.

1.3. Thesis Goal and Objectives

In order to perform a relevant contribution in the field of video based wireless sensors, a thorough evaluation of the existing methods and metrics must be performed. Consequently, the first important goal of this report is to show the performance efficient algorithms for dynamic data collection in wireless sensor networks. In order to achieve this, several objectives have to be achieved. The first one is to perform a detailed study and analysis of metrics and algorithms for video based wireless sensor networks. In addition, the drawbacks of the analyzed algorithms and metrics are important in order to achieve the proposed goal. These drawbacks are studied and the second important objective of this study is to find solutions for the problems found.

The main objective of this report is to develop software solutions for wireless video-based tracking applications. The images are taken by video cameras. Therefore two new algorithms for video based sensor networks are proposed. These algorithms are capable to save energy and in the same time to keep a certain level of coverage. Saving energy is a top research domain and there are many ways to achieve this task. The proposed algorithms differ from others because the real time component is very important. The images captured by the cameras are real data. This aspect situates the stated problem in the domain of real time algorithms for Kinetic data. A Kinetic data structure is defined [4] as any kind of continuously changing data. The process of transforming an algorithm on static data into a data structure that is valid for continuously changing (moving) data is called kinetization, according to [4]. From this perspective coverage is seen as an associated method due to the constraints. The limitations deal with real time and energy. As a consequence, the nature of the proposed algorithms has to adapt, so the constraints are satisfied.

The coverage problem in case of video based wireless sensors is close to the minimal set coverage problem, which is an NP complete one. The coverage heuristics has to be fast in order to satisfy the constraints, but also optimal in order to keep a certain level of quality. The loss in quality due to the real time constraints that have to be fulfilled is analyzed in the proposed algorithm. Also, the mentioned

limitations concerning real- time requirements and energy restrictions in balance with coverage quality are debated and satisfied in the proposed algorithm.
The overall objective regarding the new proposed algorithms is to present the performance of the proposed algorithms and their validation by simulation and by specifying their publication at different conferences and transaction. As stated, we propose two algorithms for coverage preservation while prolonging the lifetime of the network. The first algorithm computes groups of redundancy and manages sensors within those groups. The second algorithm derives from the first one, but it has no redundancy groups. The management is realized at the network level and has prediction of events. A third algorithm was also implemented based on Linear Programming, but only for comparison purposes. In order to accomplish the target mentioned, specific objectives have to be achieved. They can be classified in two areas: Proposed algorithms and simulation tool.

For the algorithms, the objectives are the following:

- Defining the context and presenting the need for developing the proposed algorithms
- Presenting the proposed algorithms
- Presenting the comparison between algorithms
- Validating the proposed algorithms by different simulations
- Validating the algorithms by using different types of sensos

For the simulation tool, Simulo, the objectives are the following:

- Presenting the main simulation capabilities of Simulo
- Presenting the features that make the simulator unique like setting the types of vehicles, the road configuration, overtaking rules, traffic behavior, setting the energy quantum
- simulate the lifetime of the network and show the improvement of the networks' lifetime when the proposed algorithms are applied

1.4. Thesis Structure

The report is structured into thirteen chapters and two parts. The first part includes the first four chapters. In the beginning a general standpoint on wireless sensor networks is presented, showing the metrics used in literature for different perspectives such as energy saving metrics, coverage metrics, etc. The first part finishes with the contribution regarding metrics by presenting the proposed metrics. The second part first presents a standpoint on current algorithms regarding coverage and sensor management in wireless sensor networks and then it presents the performance efficient algorithms for dynamic data gathering in wireless sensor networks. This part represents the main contribution of the thesis.

A brief description of the domain, the motivation and the goals of this report are presented in the first chapter. Every activity that is performed has a certain level of quality. The evaluation of the performance is estimated considering the levels of certain parameters. The video based wireless sensor networks are evaluated by some metrics. Depending on the domain where this type of networks are used, the link between quality, real-time data gathering and energy consumption can be in favor of one and not so supportive of another. The most common metrics in the field of video based wireless sensors are presented in Chapter two. One of the most used metrics for video based wireless sensor networks regards the deployment of the sensors. Deployment can be done in two ways: the sensors can be scattered (eg. from a plane) generally due to the inaccessible terrain or they can be placed at desired locations. Furthermore, the advantages and drawbacks of all discussed metric are presented, for each of the two deployment metrics. Deployment influences coverage. Next, in chapter two, coverage metrics are debated. As stated above, a good coverage usually implies more energy consumption. This issue is really important because it implies the lifetime of the network. The metrics that deal with this aspect are presented towards the end of Chapter two. In the end of this chapter, the conclusion is formulated.

The third chapter gives an overview of energy saving algorithms in wireless sensor networks. There is a considerable number of topology algorithms focused on saving energy. In video-based wireless sensors, the amount of data that is transmitted to the sink is significantly higher than in usual wireless sensor networks. Next, data aggregation algorithms are discussed. In the case of video sensors, the collected data is huge. For this reason local preprocessing of the collected data is performed. The position of the sensors is also important. If the sensors are placed in strategic positions, the number of sensors needed is low. A classic use of video based wireless sensors is traffic monitoring. Traffic surveillance is important due to prevention (eg. radars) and promptness (eg. being immediately notified of the need for intervention). Specific algorithms in this direction are presented in chapter three. Algorithms that deal with this aspect are discussed in the end of this chapter.

The metrics based on the discussion from chapter two led to the need for other metrics specific to the goal of this thesis. The proposed metrics are presented in chapter number four.

The fifth chapter represents a top view of the second part and states the general idea together with its objectives. This chapter starts the second part of the thesis that has the purpose of presenting the new proposed algorithms. First, the preliminaries and the context of the current work regarding the present traffic monitoring algorithms are presented.

Chapter number six presents in detail the first proposed algorithm. It is emphasized the fact that the proposed algorithm tries to find a balance between coverage and the lifetime of the network. The metrics proposed in chapter five are applied to this algorithm and the results are analyzed. The algorithm is analyzed from a stochastic perspective as well as from a heuristic one. Also, for a better validation, Linear Programming is used in order to compare the mathematical solution with the experimental one. From Linear Programming. LPSolve was used to implement the mathematical version of the algorithm.

Chapter seven presents the second algorithm that also deals with traffic monitoring, but it differs from the first algorithm by the new concepts implemented. For example it makes use of prediction and of events. For making an idea, events are considered to be vehicles that are driven on the same lane and the distance to its adjacent vehicles is less or equal to the minimum allowed distance. The event can dynamically form or split. The detailed explanation of the concepts and of the algorithm is presented in Chapter nine.

Chapter eight shows the performance and validation of the algorithms from Chapters six and seven. This is done by simulation and by comparing the new algorithms with the mathematical computed solution using ILP. In order to obtain real and credible simulation results real data were used in the Monte Carlo simulation. In order for the simulation to be as close to reality as possible, the mathematical foundation for the traffic simulation was realized and implemented. Traffic rules were used together with human factors like reaction time and speedy cars.

Next, Chapter nine presents the performances of the algorithms from the energy perspective showing that by using the proposed algorithms the lifetime of the network is prolonged. Prolonging the lifetime of the network is important when using wireless sensors because if the sensors used have limited amount of energy they will soon will be depleted of it. Using the new proposed algorithms, the management of the sensors will prolong considerably the lifetime of the network.

Traffic behavior issue is debated in Chapter ten. It is sown how different types of driving behavior influences the performance of the proposed algorithms with respect to coverage.

All the simulation was done using Simulo, a traffic simulator that was also implemented and represents a real contribution to this thesis. The need to develop a new simulator came from the necessity to be able to simulate both micro and macro traffic and obtain results according to the simulation on the one hand and on the other hand, to be able to simulate human behavior like speeding, overtaking on the right side only if the driver is speedy, coming back on the first lane, only if it is free, decelerating if the vehicle in front drives slower, taking into consideration the reaction time of a human being before taking a driving decision, etc. Using this simulator allowed us to implement and test the proposed algorithms using Monte Carlo simulation. The simulator is also presented in Chapter ten.

The last part is Chapter eleven, Chapter twelve and Chapter thirteen that states the conclusions, the contributions and the future work.

Part 1

Standpoint on Video Based Wireless Sensor Networks. Proposed Metrics

2. Video-Sensor Based Metrics

2.1. Abstract

This chapter presents the most used metrics in evaluating the performance of a WSN and more specific of a video WSN. Chapter number two is divided into five subchapters that deal with metrics.

First of all, the deployment of the sensors is discussed, together with the most significant metrics that deal with this aspect. Next, the coverage problem is debated in the next subchapter. Problems that appear regarding coverage are also presented. Subchapter 2.5. presents the main drawback of WSN and especially of the video WSN: energy. This chapter ends with a conclusion regarding the presented metrics.

2.2. Introduction

In a WSN a node integrates a variety of different types of sensors, depending on the area of interest. A growing demand and development is noticed in video-based sensors nodes. In addition to usual acoustic and thermal sensors, they integrate a very low powered video camera. These types of nodes collect images, process them and transmit only the potentially interesting data.

The main difference between usual nodes and video-based nodes is that normal sensors gather and process information from their own vicinity, while video cameras capture images from locations that are not in the cameras' immediate vicinity. Due to this issue, it is necessary to analyze the compatibility between the metrics applicable on regular sensors and video sensors. Another difference is that the orientation of the camera is vital in this case for the coverage issue. Different topologies may have a great impact on the efficiency of the surveillance, for example in object tracking applications.

There are some unique challenges that arise from this specific type of sensors. Compared with the usual wireless sensor networks, for video sensors imply some limitations in terms of limited energy, local computational capacity and transmission bandwidth.

In order to overcome these limitations, algorithms and techniques have been proposed. To prove their efficiency, tests in terms of metrics are applied on those algorithms. To test the effectiveness of a video based wireless sensor network, the most used metrics are those regarding the deployment and redeployment of the sensors, coverage metrics and energy saving metrics.

The parameters tested interconnect and influence one another: for example if the deployment of the nodes is not optimal and the distance between nodes is considerably big, the amount of energy needed for data traffic is higher, and as a consequence, the node will finish its amount of energy faster. These aspects are debated in [5].

This chapter presents in more detail the existing metrics for video wireless sensor networks. The challenges mentioned above are evaluated by specific metrics. The emphasis is put on deployment metrics (both scattered and willingly placed). The importance of the networks' geography is presented. Furthermore, the existing coverage metrics in literature are presented together with the motivation of having a good coverage level. In the end of the current chapter, metrics that assess the energy of a video based wireless sensor network are presented. The fact that deployment influences the amount of energy spent on transmitting data and also the fact that the better the coverage quality is, the greater the number of sensors that collect data has to be. It is relevant to mention that the lifetime of a wireless sensor network is limited, depending on the lifetime of the battery. When a sensor dies, coverage is affected. When coverage drops under a certain limit, the network is considered unuseful. Conclusion and summary of this chapter are stated at the end.

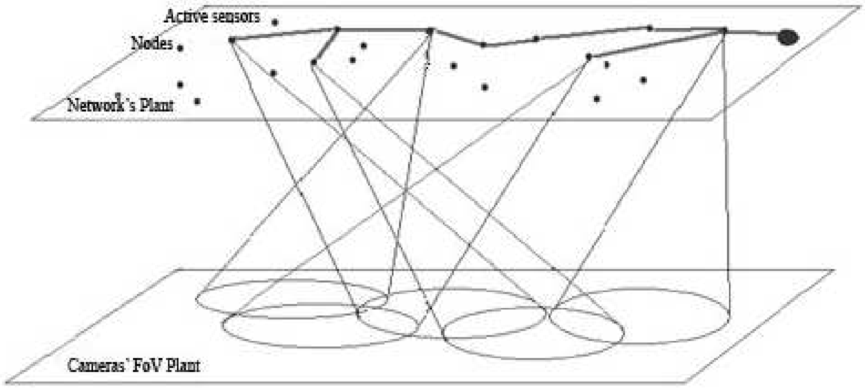
2.3. Deployment Metrics

2.3.1. Overview

Wireless sensor networks are largely used in almost every domain. These types of networks influence the scientific research in different domains and also the comfort of daily life.

Video based wireless sensor networks are composed by video sensors that have the ability to both sense (gather video information) and also to transmit that information to the central unit, if necessary.

There are two techniques of deployment and redeployment: scattering the sensors randomly [2] (e.g. from an airplane) or arranging the sensors in certain fixed positions. The literature focuses more on the algorithms concerning the second approach, but the first has its own advantages especially when the deployment area is hard accessible.



2.1. A Video-based Wireless Sensor Network deployed over the monitored area

The way the sensors are placed influences coverage. If the sensors are placed at will, an engineering topology can be precalculated and the degree of coverage is known as long as all the nodes of the network function. An issue to be considered is that in the case of video-based sensors, the data that is gathered is not in the cameras' vicinity. The field of view of video sensors is omnidirectional and this is important because not only the placement of the camera is important, but also the rotation of the video sensor. This aspect is made clear in picture 2.1. The picture is relevant to point out that in the case of video-based sensors the field of view is unidirectional, to show that the gathered information are not in the vicinity of the sensor and also to point out that the placement of the sensors is in a strong connection to the area that is covered.

2.3.2. Metrics

Most of the existing algorithms regarding the redeployment of sensor networks focus on the idea of relocation existing sensors as long as the density requirement condition for covering is accomplished. In [7], the authors present a solution to the sensor relocation problem using a maximum-flow minimum-cost algorithm. This algorithm is evaluated by measuring the improvement in network lifetime. The benefits of the proposed method are proved both mathematically and by simulations.

Another approach is to apply distributed protocols in order to provide centralized sensor deployment. This kind of redeployment is presented in [8] and [9]. These papers are focused on the usage of three algorithms VEC (VECtor based deployment), VOR (VORonoy based deployment), and Minimax to increase deployment efficiency.

In VEC, sensors that are too close to each other will be pushed away by a virtual force. The border of the network can also push sensors away. These techniques of arranging sensors are implemented in the purpose of maximizing the coverage of the monitored area.

In VOR, when a node senses the coverage gap, it will be moved towards the farthest vertex of the polygon in the Voronoi network graph. Voronoi triangulations represent the computation of the Euclidian distances between the sensors. Like in Vec, this technique of improving topology was implemented with the purpose of obtaining a better coverage in the area of interest.

Minimax algorithm is similar to VOR. Here, the virtual force will pull sensors to sparser area, but the target location differs.

Reference [10] proposes a way to achieve a maximum coverage by the usage of flip-based deployment mechanism. They assume the sensor can only flip once. The metrics applied show that the centralized algorithm maximizes the number of regions that are covered by at least one sensor node with the minimum moving cost. The methods that imply relocation of the sensors have significant drawbacks. First of all, the relocation of the sensors implies a high complexity of the solution. Second of all, there are a limited number of applications where this method can be applied. Last, these kinds of sensors are more expensive and the tradeoff between the advantages of relocation and costs is questionable.

Besides these techniques that focus on a good initial deployment, studies have been made to improve the initial deployment. One possibility in this direction is to deploy additional sensors to the ones already deployed initially. This method is called redeployment and is presented in detail in [11].

Considering the case in which sensors are randomly deployed (scattered), another problem regarding the localization of the sensors arises. In the majority of the cases, there is no prior knowledge about the location of the sensors. Many applications that use wireless sensor networks need to know the topology of the network. The solutions for this problem can be classified into fine-grained localization and coarse-grained localization. The first one measures the timing or the strength of the signal. The second one is based on the proximity from a chosen reference [12], [13], [14].

2.4. Coverage Metrics

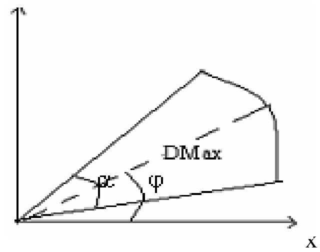
2.4.1. Overview

Considering specific applications, different types of coverage metrics are also needed. The most frequent classification found in literature concerns the network coverage and connectivity. Regarding the degree of connectivity three levels are proposed: full connectivity, partial connectivity and constrained connectivity [15]. Indeed, each approach has different performance and different cost.

Full coverage means that the each spot of the location is covered by at least one sensor. It was proven that in order to achieve a full coverage of an A size surface, the number of the sensor nodes that are needed is on the order of A ln(A).

Partial coverage is used in applications that do not demand such a stringent connectivity. Indeed, if we consider the example of meteorology, it is not mandatory to know the temperature of every spot, but it is sufficient to have 80% coverage of the area and still to provide an accurate situation of the monitored zone.

Constrained connectivity was also defined and refers to the maximum size of an area where an event can occur without being reported. A case where this kind of coverage is applied refers to the size of a fire. It can be a fire camp and in this case no action should be taken, but if the proportions of the fire are extended over a certain limit, the cameras take action by transmitting the images.

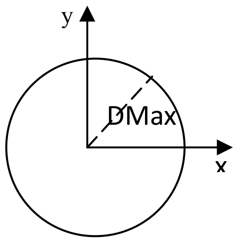


2.2. Efficient Field of View for unidirectional sensors

Besides this classification, specific matters are to be taken under discussion when talking about video-based sensor networks.

Common sensors are omni-directional. In the case of video-based sensors, the range of sensing is limited.

The cameras attached to the sensors have a field of view (FoV) that represents the area that can be observed by the sensor and also the area from which information can be gathered. An assumption that the cameras have a limited view range was made in all the considered algorithms.



2.3. Efficient sensing field for omni-directional sensors

Indeed, in order to obtain an accurate perspective, a maximum range (DMax) was established considering the video sensor resolution and the size of the smaller targets of interest. The requirement is to capture at least Np pixels of the target surface in an image. This means that the object of interest is considered to be in the field of view of the sensors only if the object is closer than the maximum range distance DMax. Then, an efficient field of view is modeled as an angle sector α of a circle with a radius DMax and centered in the node’s position. Figure 2.2 illustrates the modeling of efficient fields of view when the node orientation is an angle φ. For redundant node deployment, the field coverage is the percent of the covered surface from the total deployment surface (CS/S).

A major difference between unidirectional and omni-directional sensors is that the sensing area. Indeed, in the case of omni-directional sensors it is more general. Figure 2.3 illustrates the modeling of efficient sensing field for an omni-directional sensor.

DMax also determines the sensing area with the difference that the data is collected all around the sensor within a range smaller than DMax starting from the sensor. All the discussion from unidirectional sensors is applicable also in the case of omni-directional ones, but not the other way around.

2.4.2. Metrics

The sensor range for omnidirectional sensors means the area from which the sensor collects data. In the case of video sensors, this area is unidirectional and is called Field of view. According to [16], the Field of View (FoV) is defined as themaximum volume visible from the camera. This means that the video sensor is able to capture images that are not in its immediate vicinity, but are at a distance smaller than DMax.

In the case of video sensor networks, the three dimensional issue has been debated, due to the fact that everything can be captured in an image if it is in the FoV of the camera. This aspect involves a grater complexity of the problem. Still, [17] presents some research in this direction.

In [15] the authors analyze the cases of worst and best-case coverage in sensor networks by combining computational geometry and graph theoretic techniques, specially the Voronoi diagram, Delaunay triangulation and graph search algorithms. The paper focuses on an optimal polynomial worst and average case algorithm for coverage calculation for homogeneous isotropic sensors. The worst and best case coverage are calculated by finding the maximal breach path and the maximal support path. The maximal breach is defined as the minimum Euclidian distance from a given a path P, connecting areas I any F, to any sensor in S. The maximal support is defined the maximum Euclidian distance from path P, connecting areas I and F, to the closest sensor in S. The drawback of this algorithm is the usage of a centralized computational model. Also, the coverage depends only on the Euclidian distances from the sensors and no other conditions are considered. However, the problem of deploying sensors in order to increase coverage remains open.

Voronoi diagrams are also used by the authors of [18] propose a method a technique based on Voronoi diagram to compute an optimal path between source and destination in the presence of simple disjoint polygonal obstacles in the plane. They also take into consideration the length of the path and the smoothness of the terrain. Their algorithm is considering omnidirectional sensors and all paths are considered equally favorable.

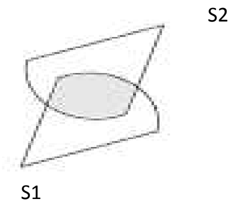
Regardless of the way the sensors are deployed (random or scattered), coverage is very important. Usually, a minimum level of coverage is estimated and the number of sensors that are deployed is established with respect to the certain coverage limit. Indeed, to accomplish an optimum coverage, when establishing that limit, the level of conectivity between sensor nodes has to be taken into consideration. The conectivity problem is debatable because there are arguments when to consider one sensor as being connected to th another sensor.

The authors of [20] present an interesting observation concerning the minimum and maximum number of neighbors that are required to provide complete redundancy and introduce simple methods to estimate the degree of redundancy without the knowledge of location or directional information. They conclude that with random sensor deployment, partial redundancy is more realistic for real applications, as complete redundancy is expensive, requiring up to eleven neighboring sensors to provide a 90 percent chance of complete conectivity in the considered area.

An important factor that influences the degree of connectivity is represented by the type of the network. If the network is a centralized one, the captured images are sent directly to the sink. From this perspective, the connectivity between nodes does not have to be so tight. On the other hand, if the network is a descentralized one, the data is transmitted from one sensor to another. In this case the sensors must have a more tight connectivity due to the fact that the data is sent through the network from one sensor to another.

Another particularity of video sensors is the fact that coverage can be affected in a dynamic manner. This happens if the field of view of that sensor is opturated by an obstacle. If this happens, that sensor becomes unuseful as long as the obstacle is preventing the sensor from capturing images. Solving this problem implies solving the solving the problem of loss coverage. The FOV loss recovery issue is difficult. Considering a simpler version of it as being the minimum set coverage for static conditions, the problem is known to be NP complete one. This aspect has been addressed by authors in [21], [22], [23].

A term that is significant when talking about coverage is redundancy. Redundancy represents an area that is covered by more than one sensor. From the sensors' perspective, redundancy implies that at least two sensors have their FOV overlapped. Figure 2.4 shows two sensors S1 and S2 that have a redundant area.



2.4. Overlapped FoV for two sensors

In order to provide a good coverage, the degree of redundancy has to be high.

A recent approach for the coverage problem is addressed in [24]. The authors propose a genetic algorithm for finding the optimal deployment in order to obtain the best coverage. They also make use of matrix definition for defining the sensing range of the sensor. In addition they also compute the Voronoi diagram and afterwards the genetic algorithm is applied. The drawback is that the genetic algorithm is not finalized and needs to be optimized more.

2.5. Energy Saving Metrics

2.5.1. Overview

Wireless sensor networks depend on the lifetime of the sensors' battery. There are wireless sensor networks that use energy harvesting techniques, but they are still in their beginnings and are not reliable yet. In the meantime, techniques that have as a purpose energy saving and metrics that evaluate those techniques are developed. A sensor has a limited amount of energy. The lifetime of a wireless sensor network represents the time in which the network can collect and transmit data at desired levels. In a WSN some nodes are more used than others. This will make those nodes to finish their energy before other nodes and become unuseful to the network. A node can be used for several tasks. It can be used to collect data, it can be used to transmit data, it can be used to process data or it can be used to perform a combination of the mentioned tasks.

If a node becomes unuseful to the network because it does not have any battery left, another node may try to perform the task the previous node did. In this way the lifetime of the network will be prolonged. Metrics that evaluate the prolonged lifetime of the network have been developed.

Each application that uses wireless sensor networks applies the technique that suits the best the purpose of the application. Prolonging the lifetime of the network can be applicable if coverage is not so important because, by using this method, in time the coverage quality will drop until the network will become out of use.

In the case of video-based wireless sensor networks, the energy problem affects even more the performance of the network. In the case of video sensors, the data they collect are images. In some applications like fire prevention, when the events of interest happen rarely, not all data has to be transmitted to the sink. There for before the information is communicated to the central unit, local preprocessing is performed.

One of the most relevant general metric in wireless networks is the remaining battery power of nodes. Experimental measurements indicate that the communication cost in wireless ad hoc networks can be two orders of magnitude higher than computation costs in terms of consumed power [25].

2.5.2. Metrics

Establishing the amount of energy spent in a wireless network can be difficult, especially when the communication varies dynamically with respect to the surrounding stimuli. In [26] the authors try to consider a scenario in which a wireless sensor network is formed by randomly deploying n sensors to measure some spatial function over a field, with the objective of computing a function of the measurements and communicating it to an operator station. They establish scaling laws for the computation time and energy expenditure for one-time maximum computation. They show that for an optimal algorithm, the computation time and energy expenditure scale, as $\theta\left(\sqrt{\frac{n}{\log n}}\right)$, whereas the energy expended scales as $\theta\left(n\sqrt{\frac{n}{\log n}}\right)$, n being the number of sensors, $n \rightarrow \infty$.

Energy saving metrics evaluate not only the battery level, but also the performance of the network. There are applications where every frame is important. Among these applications are the traffic surveillance ones. In these applications, the energy consumption is greater than the case of fire surveillance, mentioned above. The data that have to be transported to the sink are greater, so the traffic performed is significantly higher. Moreover, the constraints concerning real time have to be respected. In this direction, the delay of the wireless network is evaluated. A metric that estimates the balance between the capacity of making real time decision and the remaining amount of energy is frequently used in these types of applications. The delay is significant in video WSN because it is important that the data reaches the sink in time to make immediate decision if necessary. Reference [27] presents two main factors that influence latency: physical distance in over which the data has to be transmitted and the number of hops.

The author of [27] shows that the power consumption can be reduced when the node density increases in a wireless sensor network. In addition to this, the paper also states that the networks' capacity grows when additional nodes are deployed. The increase of the network is due directly to a reduction in the interface between transmissions when the transceivers are operating at reduced power. In the paper this conclusion was proved analytically.

Reference [28] presents some metrics for wireless sensor networks that are applicable, from the energy perspective to video based wireless sensor networks, as well. They adjust the transmission protocol in order to keep a certain link quality. They introduced the link quality indicator and the received signal strength indicator as metrics.

Authors of [29] propose a tradeoff Energy Savings and Source-to-Sink Delay in Data Dissemination for Wireless Sensor Networks. In order to test their solution, they apply two metrics: energy consumption and the delay from source to sink. They prove that the nodes that are deployed closely to each other and can form a path from source to sink consume the minimum amount of energy. The disadvantage is that certain sensors will be used frequently and their energy will be consumed first. Adopting this technique, after the optimal sensors from this perspective, will die, the network will become more energy consuming than in the case of normal path usage due to the disadvantaged remaining paths.

2.6. Conclusion

Wireless sensor networks are used at a large scale in different applications. The performance of a wireless sensor network is performed by applying different metrics. Not all the metrics that are used for usual wireless sensor networks can be applied in the case of video based wireless sensor networks because in the case of video WSN, the FoV is unidirectional.

Analyzing the existing metrics led to the conclusion that there has to be a balance between the efficiency of the network and the amount of consumed resources. The efficiency of the network, in most of the cases means, first of all a good deployment that leads to obtaining good coverage, a prompt response that means real time abilities of the network where latency is reduced as much as possible. All the mentioned metrics were discussed in this chapter. The levels of the specific parameters are influenced by the needed resources.

One of the main drawbacks these types of networks have is the limited amount of energy. There are different techniques that try to overcome this limitation. Energy represents a problem for all WSN, but in the case of video WSN, this problem is even greater due to two main factors: local data processing must be performed and the collected data are images. Each of the two is more energy consuming than a usual WSN. The metrics applied to evaluate the performance of energy saving techniques also evaluate other parameters that lead to more consumed energy. One example is the latency of the network.

It has been proved that the amount of energy that is used by the network is directly proportional to the quality of the network. There are two directions when talking about energy saving. One aspect is to find methods that save energy, but in the same time to be able to maintain a certain level of efficiency of the wireless network. Another aspect regards prolonging the lifetime of the network. The metric applied in this situation measures the time the networks' lifetime was prolonged.

3. Performance Efficient Algorithms in Wireless Sensor Networks: Routing, Aggregation and Deployment

3.1. Abstract

This chapter presents in more detail the existing approaches for the problem of limited energy in wireless sensor networks. Mainly the specific algorithms for this purpose can be classified in three directions: topology algorithms, data aggregation algorithms and deployment algorithms.

Topology algorithms are important due to the fact that the data has to respect the real time constraints. The amount of information that is transmitted in the case of video sensors is significantly higher than the data collected in the case of usual sensors. Furthermore, if the traffic is done between sensors that are placed at a considerable distance, the amount of energy grows in this case.

Data aggregation techniques for saving energy are the most researched and for a better efficiency, can be combined with a good topology strategy.

Deployment plays a big role in topology. If the sensors are scattered, the number of deployed sensors has to be higher.

The end of the chapter concludes about the energy efficient methods.

3.2. Introduction

Minimizing energy consumption in order to prolong the lifetime of the network is a major challenge in the domain of wireless sensor networks. Researchers are exploring advanced techniques looking for solutions both in the hardware and software domains for saving energy in WSN.

The hardware approaches generally refer to the capability of the nodes to be turned off completely. This implies the usage of a topology control algorithm that can reconfigure the connections between nodes in order to avoid data transmission through the turned off nods, but still to keep the network connected. A different approach is to keep certain nodes off in order to save energy. Others have studied the possibility of periodically checking whether a node should become a coordinator or to enter a standby mode. All these techniques have to take into consideration the fact that in the majority of applications, decision making is really important from the real time perspective. If a sensor is in stand-by mode and at a certain time, that sensor becomes the optimal one to be turned on, the mode changing has to be done in real time. Otherwise, the sensor would not be able to accomplish the task that it was chosen to do.

The fact that data transmission is very expensive in terms of energy use was shown in [25]. It is obvious that reducing transmission costs would reduce total energy consumption. Transmission is improved by finding the best path for data transmission. It is debatable what best path means. In literature several approaches that present this theme can be found. A possible method is to find the shortest path. Another method is to find the path that has the minimum number of hops because each hop introduces a certain delay in data transmission. To respect the real time transmission another topology solution is to find the most reliable path that avoids any traffic collision. Topology methods combined with the hardware methods mentioned above have to consider the fact that if a node is completely turned off, it can't perform any data transmission, but the energy saving amount is bigger. Another possibility is to turn only the camera off, so the node is able to perform traffic.

Deployment is another option for saving energy in a wireless network. This technique can be used mostly for the willingly deployed sensors that have a known topology. In this case the degree of coverage that will be obtained as well as the degree of redundancy is known. The sensors can be deployed in an optimal manner. In the case of scattered sensors, the deployment is random and to obtain certain coverage, a bigger number of sensors have to be deployed. If the data is gathered from more sensors, the amount of energy used is higher.

In this chapter the approaches regarding energy saving techniques are presented. In the first part topology algorithms are discussed. Next, data aggregation methods are showed. Towards the end of the chapter, the role of deployment in energy saving is presented. This chapter ends with conclusion and summary.

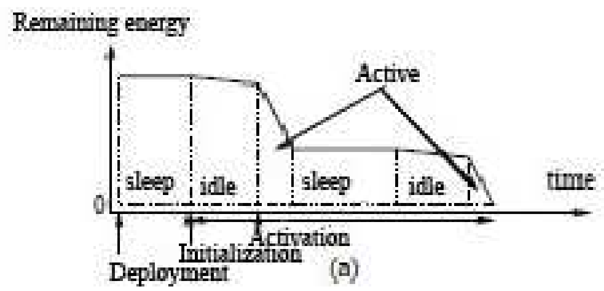
3.3. Topology Algorithms

To maximize the overall network lifetime, in [30] an algorithm that minimizes the routing energy by choosing paths through a multi-hop ad-hoc network is presented. In this power saving method, nodes adjust their transmission power levels and select routes to optimize performance. Energy saving by routing methods is also discussed in [31], where a technique based on Directed Diffusion is proposed. The authors describe Directed Diffusion routing and illustrate one instantiation of this paradigm for sensor query dissemination and processing. They show that using Directed Diffusion one can realize robust multi-path delivery, empirically adapt to a small subset of network paths, and achieve significant energy savings when intermediate nodes aggregate responses to queries.

An interesting work on energy saving in WSN by routing approaches is presented in [32]. It presents an analysis of the lifetime extent of a wireless sensor networks that employ periodic sensing. Lower and upper bounds on the network lifetime are derived, and corresponding routing algorithms leading to these bounds are presented. For large sensor networks the upper and the lower bounds on the network lifetime are relatively close (less than a few percents), leading thus to the conclusion that for such sensor networks the choice of the routing protocol is largely irrelevant for maximizing the network lifetime, as long as some form of shortest paths are followed. Simulations are used to validate the theoretical results. Other approaches discuss the possibility of minimizing the energy consumed for each message [33], [34], and [35]. This metric might unnecessarily overload some nodes causing them to die prematurely. Minimizing the variance in the power level of each node [36] is another possibility that saves energy. Minimizing the maximum energy drain of any node is a solution discussed in [37] and [38].

Another promising solution to save energy in video-based networks is presented in [39]. The authors propose a coordination algorithm for topology maintenance. The algorithm adaptively elects node coordinators from all the nodes in the network, and rotates them in time. Coordinators stay awake and perform multi-hop packet routing within the ad hoc network, while other nodes remain in power saving mode and periodically check if they should wake up and become coordinators.

In reference [40] the authors introduce two variables: base stations that have flexible positions within a certain range and application nodes that have the role to receive data from the sensor nodes. Two approaches are studied in this article. One refers to the location of the sensors and the other one considers parallel relay routes in order to obtain an optimal allocation. The best route is chosen for data transmission. In the study performed in this paper, the authors realized a graphic that shows the link between the amount of the remaining energy and time. During the experiments the nodes changed their status from sleep to idle, to active, to sleep again and then to idle again, etc. Figure 3.1 illustrates this connection.



3.1. Activation modes. Discrete model [40]

Another approach presented in [41] proposes two techniques for energy saving. The first one implies that each node remembers its past activity and if in a chosen time interval, no activity was detected, the node goes to sleep. The sleeping period is a fixed amount of time. After waking up, the node again memorizes its activity. The second approach is based on voting to decide if a node should go to sleep or not. The sleeping period is also a fixed one. These approaches are linked to topology due to the fact that the algorithm is distributed. This means that each sensor takes decision by its own, including the decision of message transmission. There are two problems that arise. One question would be if the local optimum is the same with the local optimum. The other question is how much information is lost during the fixed time in which the sensors are in sleeping mode.

3.4. Data Aggregation Algorithms

In [42] each node periodically decides whether to sleep or stay awake as a coordinator. A node decides to be a volunteer to be a coordinator if it discovers that two of its neighbors cannot communicate with each other directly or through an existing coordinator. Similar to this approach, [43] presents an algorithm that for each node maintains a count of the number of nodes within radio range, obtained by listening to transmissions on the channel. A node switches between sleeping and listening, with randomized sleep time proportional to the number of nearby nodes. As a consequence, the number of listening nodes is quite constant. The difference between the two algorithms is that Span never keeps a node awake unless it is essential to connect two neighbors.

Reference [44] proposes an energy-efficient optimisation approach to achieve tracking accuracy constrained by energy consumption. It enables reorganization of wireless sensor networks and includes three phases, which are related to prediction, localization and recovery. The first phase implies a particle filter algorithm on the sink to forecast the future movement of the target. Then, the most energy efficient sensor nodes are awakened to locate the target. Energy efficiency is calculated as the ratio of mutual information to energy consumption. The recovery phase is performed when the target is missed because of the incorrect predicted target location and implies a genetic-algorithm-based mechanism.

The quality of images is a major reason for consuming processing power in video-based WSN. This is directly influenced by resolution and frame rates. A novel algorithm that implies image-processing techniques is presented in [45]. It aims to reduce the workload of individual sensors. Given the limited resources of sensor nodes, the approach exploits the redundancy among nodes by partitioning the sensing task to highly correlated sensors. For an object of interest, each sensor only captures and delivers a fraction of the entire scene. Then, the partial images are fused together for reconstructing the image. Experiments show that this approach achieves promising results. The authors offer detailed discussions about the effect of the variance for different algorithm parameters.

The work in [46] proposes an energy-efficient optimization approach to achieve tracking accuracy constrained by energy consumption. It enables reorganization of a wireless sensor networks, and includes three phases, prediction, localization, and recovery. The first phase uses a particle filter algorithm on the sink to forecast the future movement of the target. Then, the most energy efficient sensor nodes are awakened up to locate the target. Energy efficiency is calculated as the ratio of mutual information to energy consumption. The recovery phase is performed when the target is missed because of incorrect prediction of the target location, and is based on genetic algorithms.

A major factor that consumes processing power in video-based WSN is the quality of images. This is directly influenced by resolution and frame rates. A novel algorithm that implies image-processing techniques is presented in [47]. Its goal is to reduce the workload for individual sensors. Given the severe resource constraints on individual sensor nodes, their approach is to employ the redundancy among sensor nodes by partitioning the sensing task among highly correlated sensors. For an object of interest each sensor only needs to capture and deliver a fraction of the scene. Then these partial images will be fused in order to reconstruct the whole image. The experimental results show that this approach can achieve satisfactory results and the authors give detailed discussions on the effects of variance of different algorithm parameters.

A solution that uses a hardware platform that allows a node to be put in stand-by, but still to allow traffic is presented in [48]. This means that stand-by nodes do not gather information, but are able to route messages. This kind of node allows the network aspects and image sensing to be separately treated.

Another solution for energy saving is presented in [49]. The method addresses the way the sensing task is partitioned among sensors. It proposes an image fusion algorithms based on epipolar line constraint to fuse the received partial images at the sink.

The approach in [50] addresses shadow-induced errors in image acquisition. It detects and suppresses shadows using the color ratio between lit and shadow pixels.

A novel traffic congestion monitoring method is proposed in [51]. The method models cars as agents that deposit pheromone at virtual places. Pheromone evaporates and propagates following a modified version of the state transition model for digital pheromone. A car predicts the traffic on the road ahead from the information provided by the preceding cars.

The system in [52] uses video cameras mounted on buses to dynamically monitor the traffic conditions along a traffic corridor.

3.5. Deployment Algorithms

In order to describe WSN deployment efficiency, several metrics have been proposed [53]. Mainly the article treats the possibility of some metrics that are used on regular sensors, to be used for video sensors, as well.

The authors of [54] prove experimentally that from the image coding perspective, the distance between sensors influences the amount of consumed energy. In a centralized network, due to the fact that the intra encoder and decoder have almost the same complexity, so coding is suitable on both cases when the transmitter as well as the receiver is energy constrained. If the coding is applied in a distributed manner, video coding is suitable in WSN deployments where only the transmitter is energy constrained.

Reference [55] proves analytically that in the case of video wireless sensor networks, from the energy efficiency point of view, it is preferable to have a high density network. In this case, the transmission would be much faster and easier, having more available routing possibilities. This way, a considerable amount of energy is saved, the real time constraints are fulfilled and the obtained coverage is high. The authors of this paper analyzed, as well, the drawbacks. The main disadvantage is cost, but they conclude that despite the investment, this method proves its efficiency.

3.6. Conclusion

While existing methods offer valuable contributions, there are still many challenges to be addressed, like the high computational complexity of video processing, difficult modeling and prediction due to a large variety of situations, and the presence of disturbances (like moving obstacles) that perturb the efficient acquiring of images.

Sensors are largely used. Wireless video sensors are still at the beginning. The consumed amount of energy is significantly higher than in the case of usual sensors. Furthermore a wireless network is useful if it reaches the proposed goals. Generally the performance of a network is quantified, as discussed in chapter 2, in the quality of coverage, real time response and efficient energy consumption. Unfortunately, as seen in this chapter, there always exists a tradeoff between the performance of a network and the consumed amount of resources. The better the performances are, the more the amount of resources needed to accomplish the task grows. There are a many proposed solutions to this problem. Hardware solutions, as well as software are tried. The trend in this area is the use of harvesting energy. The existing applications with video sensors that use harvesting energy are not wide spread yet. If the energy limitation will not exist anymore, the problem that will still persist will be the real time constraints.

4. Proposed Metrics

4.1. Abstract

This chapter presents the analysis that was performed mostly in terms of coverage for both scattered and panned deployment.

First, the argument regarding the need of metrics in video WSN is presented together with the motivation. Next, the proposed metrics are presented. In the beginning the estimation of uncovered surfaces and paths is done. Furthermore different metrics are applied to establish the influence of the deployment method upon the degree of coverage. The number of continuous uncovered surfaces is counted. Also the number of continuous uncovered crossing paths is computed. A conclusion for these metrics was drawn.

Next, the benefits of an algorithm proposed for saving energy are analyzed. The metrics presented in the first part of this chapter are applied in order to obtain a comparison between the coverage when no scheme for sensor management is applied and the case when an algorithm is applied. Simulation results and a conclusion for this section are presented.

In the end of this chapter a general conclusion is drawn based on the simulation results.

4.2. The Need for Metrics in Video WSN

Wireless sensor networks are largely used and they are still developing. They are used in a lot of domains from health care, agriculture, military applications to weather forecast and traffic monitoring. The contribution of WSN in the mentioned domains is obvious. This is one of the reasons why the area of WSN is so intensely researched in order to overcome and find a solution to their challenges.

The major limitations of wireless sensors are the limited amount of energy, the real time information processing, the reliable data communication and the link between the sensors and the users' interface. All these issues represent variables that cannot be fully predicted with respect to their reliability. As a consequence, the functionality of the wireless network is, in a certain amount, a probabilistic one.

In order to compensate the trust, the wireless network is design to be specific for the applications that uses it. This way, the designer of the network fins a tradeoff between all the variables and improves the ones that are most important for the application. Due to the unknown factors mentioned above, another difficulty resides in integrating the network into a bigger application.

When talking about wireless video sensor networks, the problem gets more complicated because in addition to the usual components of a sensor, a wireless sensor has a little camera that is attached to the device, camera that captures images, and compresses them to a desired format. If necessary, at the node level, a preprocessing is performed, as well. This is done in order to transmit only the important data and, this way, to ease the amount of data traffic.

In the case of wireless video sensors, all the mentioned challenges stand, besides the stated limitations of a regular node, some uncertainties are amplified and new ones appear. Video cameras are more energy meaning, even though the camera has reduced proportions, so the energy dilemma, in the case of video sensors, sometimes gets to be a real problem. Video sensors consume more energy in two more directions: one is the camera and the other one is the fact that, in most of the cases, local preprocessing is required. Besides the energy issue, video sensors bring along the unidirectional field of view. The data are not collected from the sensor's vicinity and the process of collecting images can be disturbed by the interference of unpredicted obstacles in the field of view of a sensor. This may lead to unreliable collected data and coverage deficiency. Furthermore, the fact that in the majority of video sensors, local data preprocessing is performed, the real time constraints may be affected.

When dealing with a system, the main desire is to have trust in the reliability of the system. Unfortunately, until now, the domain of wireless sensor networks and especially that of video wireless sensor networks cannot be fully predictable systems due to the variables discussed above.

The need to have a system, as predictable as possible, metrics have been introduced and applied to wireless networks, and to systems, in general. In the case of wireless sensor networks there are no result standards due to the mentioned variables. Still, an evaluation of the networks' performance must be done. This becomes the task of the metrics. Metrics are applied to the network and they measure the results of the network for specific issues like the amount of energy used in a certain amount of time, or the coverage level of the network, etc.

It is important to mention that, generally, in the case of wireless sensors, metrics are not applied to confirm that the network behaved as expected, but to measure the level of unpredictable variables. In this way, the network's parameters can be adjusted in order to obtain a system that is specific to the desired application.

4.3. Motivation

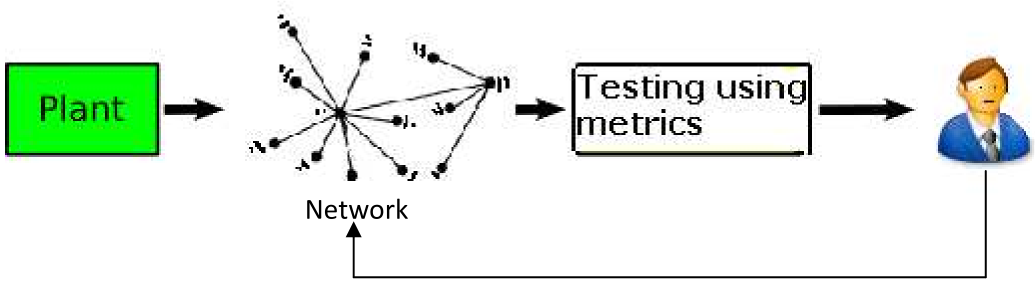
Video wireless sensor networks are now wildly used. Even though there are no standards regarding the discussed variables, the users of the networks must have confidence in the system they use.

For this report I proposed an algorithm for video sensors management that finds a balance between energy, coverage and real time constraints. This algorithm was tested at each step. The metrics used were not a standalone purpose, but came as a necessity.

The process of bringing the algorithm form the state of an idea to implementation and testing was complex and demanded a circular progress: from algorithm implementation to testing, again adjusting the algorithm, again testing, so on.

The motivation of applying new metrics came from the fact that metrics for video sensors are harder to find in literature because this domain is in its early age. Most of the metrics are suitable for omnidirectional sensors. Due to the restricted sensing area, metrics have to be restricted, as well.

The circular process described is illustrated in Figure 4.1.



4.1. The cycle of adjusting the parameters of a wireless sensor networks

The proposed metrics were applied on different test scenarios, with different purposes: for example in the case of scattered sensors, coverage levels are analyzed.

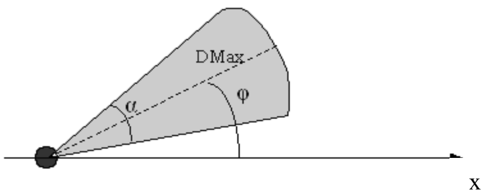
4.4. Metrics

4.4.1. Metrics for Estimation of Uncovered Surfaces and Paths

The majority of the algorithms are designed for specific problems. Despite of common applications' demands, most of the analyzed related work refers to the case where the cameras are located above the plan of surveillance and the sensors are omnidirectional. To overcome this drawback this study focuses on the performance of the covering metrics applied to video-based sensors that form a network located on the ground, in the surveillance plan with specific properties.

These algorithms present a particular relevance in the field of object tracking, intrusion detection, and general surveillance. Together with proposed metrics they are applicable for all types of deployment discussed above, but in this chapter the analysis are done for scattering deployment. Considering these specific cases, the metrics and algorithms proposed in the present cases are unique due to the conditions imposed.

For testing, the conditions assumed were that all the cameras are identical from perspective of the resolution and the view angle. In addition, a new approach was adopted. In this work we consider that the cameras have a limited view range. In order to obtain an accurate application perspective, a maximum range (DMax) was established considering video sensor resolution and the size of the smaller interesting target. A condition is to have an extent of minimum n pixels of target on image. This means that the object of interest is considered in the field of view of the sensors only if the object is closer than the maximum range distance DMax. In literature the Field of View (FoV) is defined as the maximum volume visible from the camera [53].

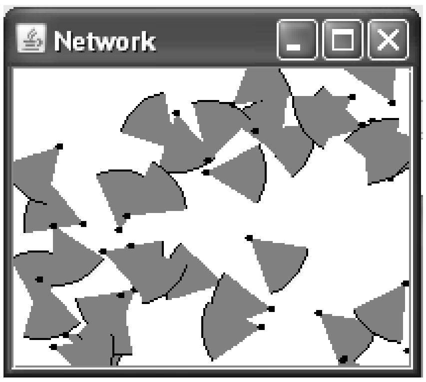


4.2. Efficient Field of View (EFoV) and camera orientation for a video sensor

The camera therefore is able to capture images of distant areas and objects that appear within the camera's depth of field. In our work we consider the FoV as the intersection of that volume with the ground plane. We also define the Efficient Field of View (EFoV) as an α angle sector of a circle with a radius of DMax and centered in node position. Figure 4.2 illustrates the EfoV considering also node orientation denoted by an angle φ. Another remark is that the considered sensor nodes are not mobile. They are scattered randomly and remain fixed in the initial positions.

In most of the cases, the sensors do not cover the entire surface. Figure 4.3 presents a case of a partial coverage.

We propose two metrics for determining the efficiency of the deployment. The first metric denotes the covered surface (CS/S) from the total deployment surface. Even if this is a common approach, the conclusions drawn are relevant for most applications.



4.3. An illustration of coverage on a rectangular monitored area

We also propose an algorithm that calculates this metric It uses a discrete representation of FoVs projection on monitored aria A and achieves an O(n2) complexity.

One relevant test done using this algorithm is the CS/S variance on a fixed size surface with respect to a linear increasing network size. Another aspect that was analyzed is the variance of CS/S on several random deployments of a fixed size network. The results of these tests are presented in the next section.

To increase the performance of the first metric in case of intrusion detection applications, a second one is introduced. Considering the case of intrusion detection, an important issue is to determine the size of the Maximum Continuous Uncovered Surfaces (MCUS) on monitored area. This is relevant to realize how much a target can move in the area without being noticed by the network.

Associated with MCUS we also consider the deployment homogeneity expressed by the total Number of Continuous Uncovered Surfaces (NCUS).

Finally, the particular case in which the target can traverse the surface between two borders is analyzed. This situation is captured by a fourth metric named Number of Crossing Paths (NCP). The NCP will count the number of different uncovered paths that cross the network. Each path will start from a different continuous uncovered surface.

4.4.2. Experiments

4.4.2.1. Abstract

In analysis we presented metrics that allow the analysis of deployment performances for a wireless sensor network. The defined metrics are the covered surface from the total deployment surface (CS/S), the maximum continuous uncovered surfaces (MCUS) on monitored area, the number of continuous uncovered surfaces (NCUS) and the number of crossing paths (NCP). With these metrics, relevant tests regarding the covered surface variance on 100 network deployments was performed. They include the variance of surface covering when growing the number of nodes, the number of continuous uncovered area, the maximum area and the possibility for a target to travel undiscovered from one border of area to another.

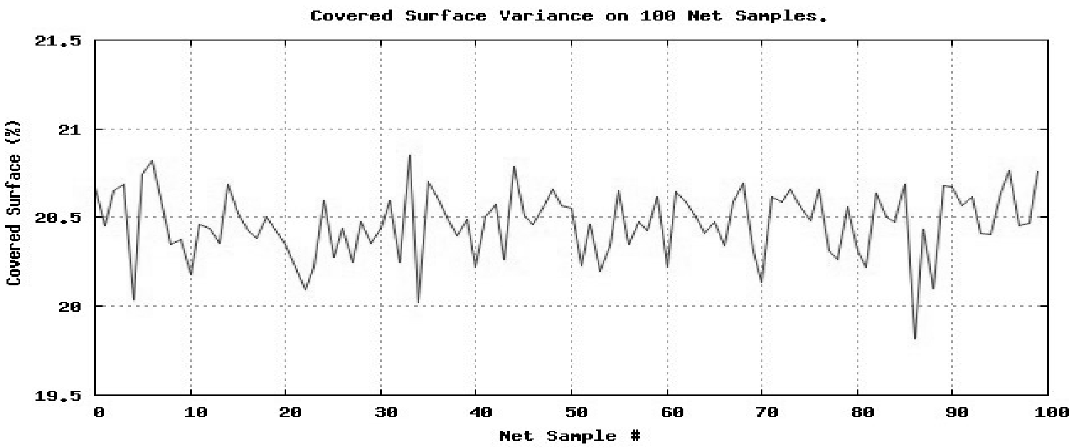
4.4.2.2. Experimentation Platform

All algorithms that perform metrics computation have been implemented and used in several studies as stand-alone Java packages.

Studied topologies were generated using a uniform random distribution provided by a standard java library class java.util.Random. We consider only homogenous networks. All camera nodes have same characteristics as a video resolution of 160x120 pixels and a view angle of 60 degree. DMax was estimated using a Trendnet IP-400W wireless surveillance camera, considering an adult person as a smallest target. Using that heuristics, DMax was set to 30 m. As a deployment area we consider a plain rectangular 1.000x1.000 m2 field. No obstacles or hard environmental conditions were considered.

4.4.2.3. Results for CS/S Metric

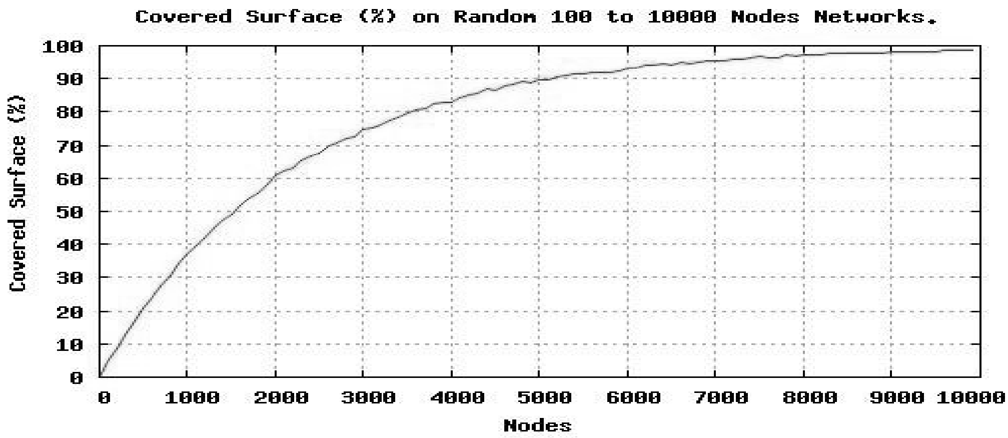
In this section, we present several experiments using CS/S metric and try to provide analysis of results. Figure 4.4. shows the magnitude of CS/S dispersion for 100 uniform random deployments of a 500 nodes network on a one km2 plain field. The result shows a maximum dispersion less than 1%.



4.4. CS/S dispersion on 100 random deployments of a 500 nodes network on a 1.000x1.000 m2 monitored area

4.4.2.4. Results for MCUS and NCUS Metrics

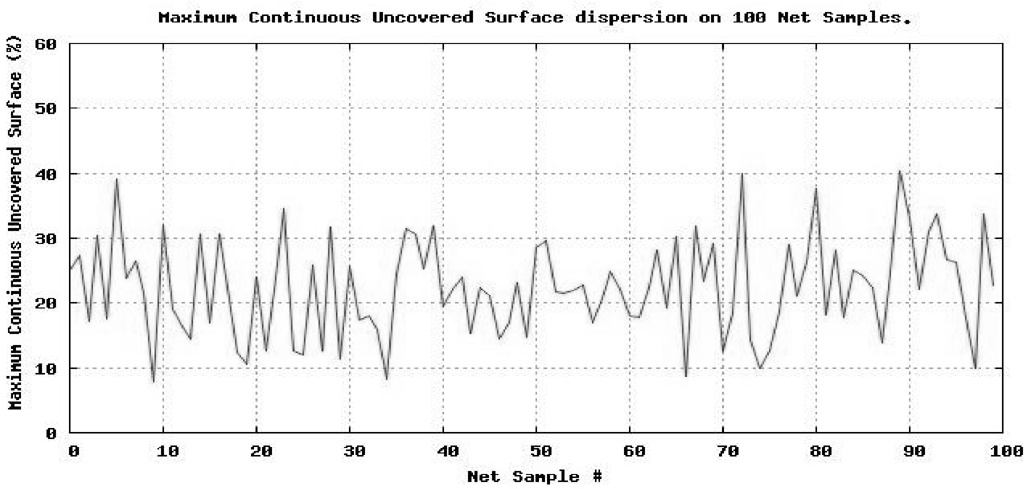
This section presents several experiments using MCUS metric in conjunction with NCUS.



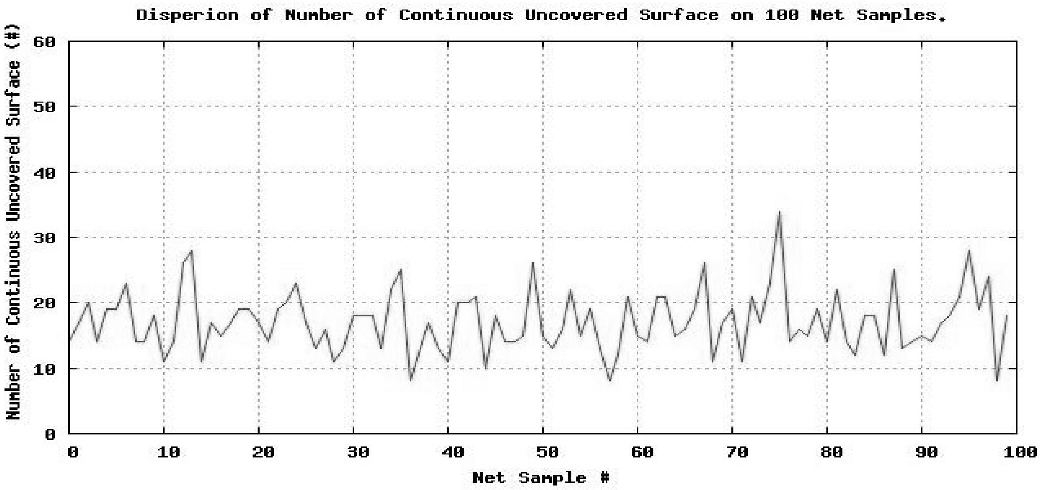
4.5. CS/S variance on deployments of networks having sizes between 0 and 10000 nodes on a 1000x1000 m2 monitored area

The result presented in Figure 4.6. demonstrates a higher dispersion (>25%) than in the case of CS/S.

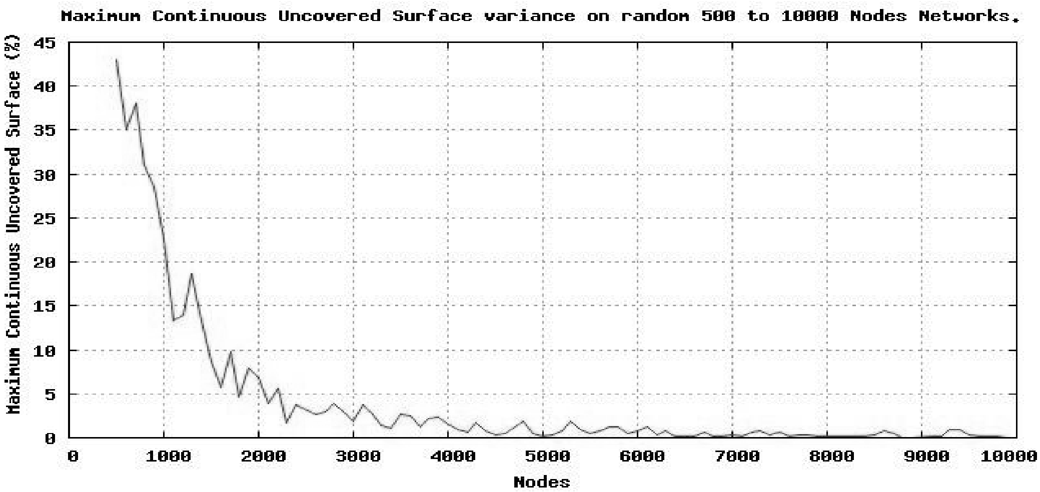
Indeed, if we consider also an associated value represented by number of disjunctive continuous uncovered surfaces, an important dispersion in absolute value will be noticed, as seen in Figure 4.7. This suggests a special attention in applying these metrics for redeployment.



4.6. MCUS dispersion on 100 random deployments of a 500 nodes network on a 1.000x1.000 m2 monitored area



4.7. NCUS dispersion on 100 random deployments of a 500 nodes network on a 1.000x1.000 m2 monitored area

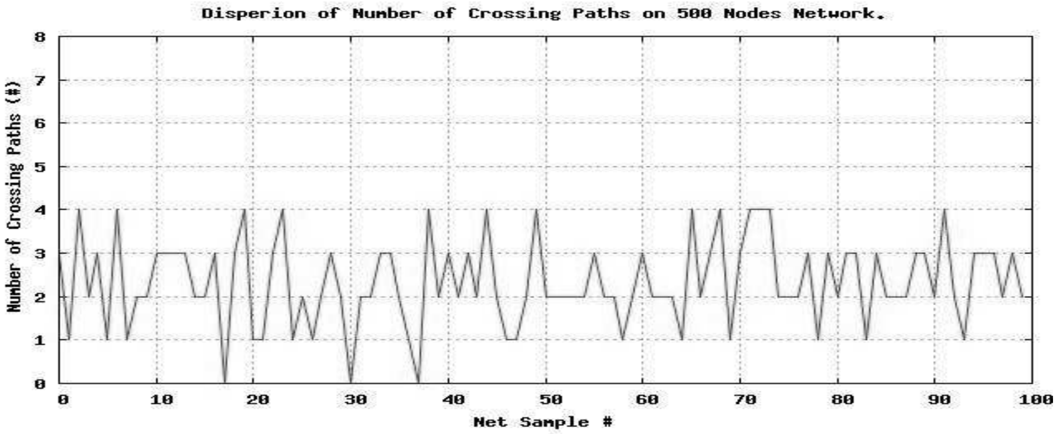


4.8. MCUS variance on deployments of networks having sizes between 0 and 10000 nodes on a 1000x1000 m2 monitored area

Figure 4.8. shows an expected variance of MCUS when the network size is increased but the influence of high dispersion could be obviously noticed.

4.4.2.5. Evaluation of NCP Metric

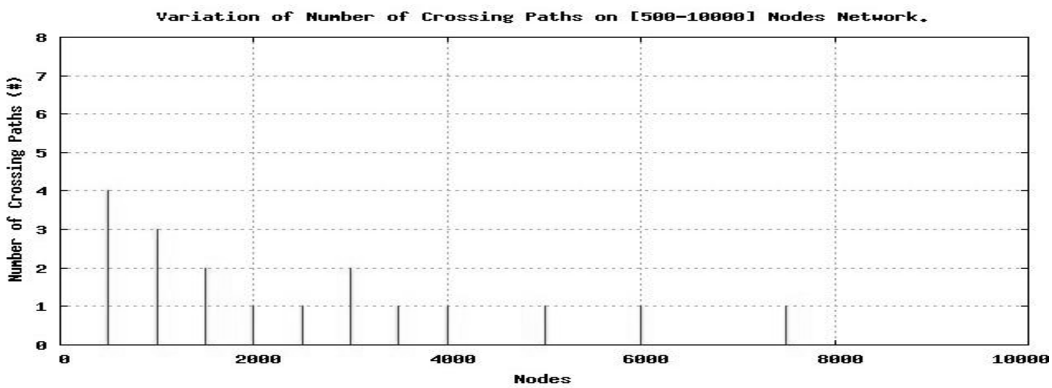
The NCP metric expresses the number of disjunctive uncovered paths that traverse the monitored area. A path is defined as a line that starts from one border of the area and ends to a different one. Two paths are considered disjunctive if they belong to different continuous uncovered surfaces.



4.9. NCP dispersion on 100 random deployments of a 500 nodes network on a 1.000x1.000 m2 monitored area

Figure 4.9. depicts the dispersion of NCP for 100 random deployments of a 500 nodes network.

The graph presented in Figure 4.10. is affected by significant dispersion but still suggest a decreasing trend of NCP. The result allows us to get an estimation of minimum number of nodes to achieve no uncovered crossing path along network. However, in case of general shape area the definition of NCP is less significant.



4.10. NCP dispersion on 100 random deployments of a 500 nodes network on a 1.000x1.000 m2 monitored area

4.4.3. Metrics for determining the influence of an algorithm for energy saving applied in the case of deployed and scattered sensors

4.4.3.1. Abstract

This subchapter realizes a comparison between the variance of coverage when an algorithm for sensors' management is applied. The results show that there is not such a big difference with respect to coverage between the random deployed sensors and the planned deployed ones. A difference can be observed between the effects of the strategies applied. Strategy2 that has a node management algorithm offers a better coverage than Strategy1. Due to the fact that redundant nodes are turned off, also energy is saved and the lifetime of the network is prolonged.

4.4.3.2. Introduction

For each case of intersection, the overlapping area was calculated and a decision was taken with respect to the degree of redundancy. Cases where obstacles (cars) were covering the FoV of a sensor determine us to find a smart algorithm that does an intelligent sensor management. These cases were analyzed when sensors were scattered and also in the case where sensors were placed at will.

As mentioned earlier, the field coverage is the percent of the covered surface from the total deployment surface (CS/S).

The algorithms that are analyzed in this chapter have an essential factor the percent of the covered surface. One relevant test done using this algorithm is the CS/S variance on a fixed size surface with respect to a linear increasing network size. Another aspect that was analyzed is the variance of CS/S on several random deployments of a fixed size network. The results of these tests are presented in the next section. Two metrics for determining the efficiency of the deployment were proposed. One metric obtained with this algorithm denotes the covered surface (CS/S) from the total deployment surface. Even if this is a common approach, the conclusions drawn are relevant for most applications. An algorithm in this purpose was implemented. It uses a discrete representation of FoVs projection on monitored aria A and achieves an O(n2) complexity. The algorithm determines the degree of redundancy and turns off the nodes that have CS/S greater than 70 percent. These nodes can be turned on again when necessary.

On the other hand, the case where sensors are placed at will was considered the one for traffic monitoring. For this case the algorithm also calculates the percentage of CS/S and turns off the most redundant node, but in its place, an optimum node for coverage is turned on. The idea was that the covered shouldn't go less than a limit imposed.

In this chapter we analyze the redundancy degree of nodes that are scattered and of nodes that are placed at will and we also analyze how the presented metrics differ in each of the cases.

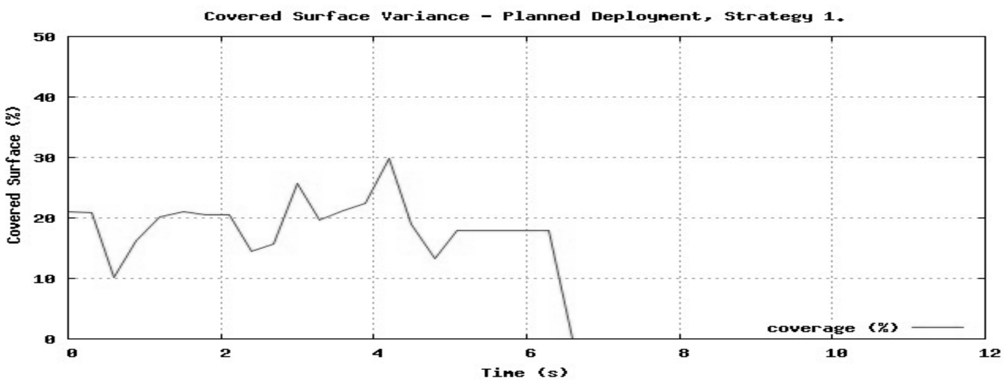
4.4.3.3. Simulation results

The proposed method was implemented as a Java program, and simulated on a PC desktop computer. Experiments studied the coverage variation when managing redundant nodes and in the presence of moving obstacles and for different traffic scenarios.

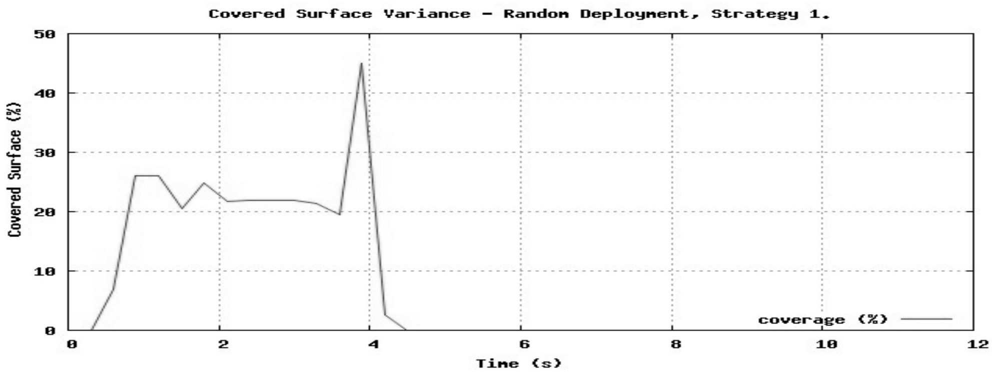
Figures 4.11., 4.12., 4.13. and 4.14. give more insight about the covered FOV areas in the case of random deployment and also with a planned deployment. These cases were simulated using two algorithms, named Strategy1 and Strategy2. Strategy1 calculates the coverage variation when obstacles appear in the FOV of the sensors, without changing the status of the sensors. If they were off, they remain off and also, if they were on, they continue to be on even if their FOV is covered by the obstacle. Strategy2 contains an algorithm that manages the status of the redundant sensors. In the algorithm, a sensor was considered to be redundant with another sensor if their FOV overlapped n a proportion larger than 70%. The strategy calculated the redundant sensors for the sensor that had its FOV covered by the obstacle and turned the sensor with the covered FOV off, turning on, instead the most significant sensor. The most significant sensor was considered the most redundant sensor with the sensor that was turned off.

For each of the strategy, tests were done in the cases of random deployment and planned deployment. The camera model used the following parameters (see Figure 4.2.): DMAX was 30 meters, and was 40 degrees. The distance between the camera sensors and the monitored route was set between 2 and 6 meters. The simulated time was 8 seconds and the simulation step was 0.3seconds. In all the test cases, 5 sensors were used.

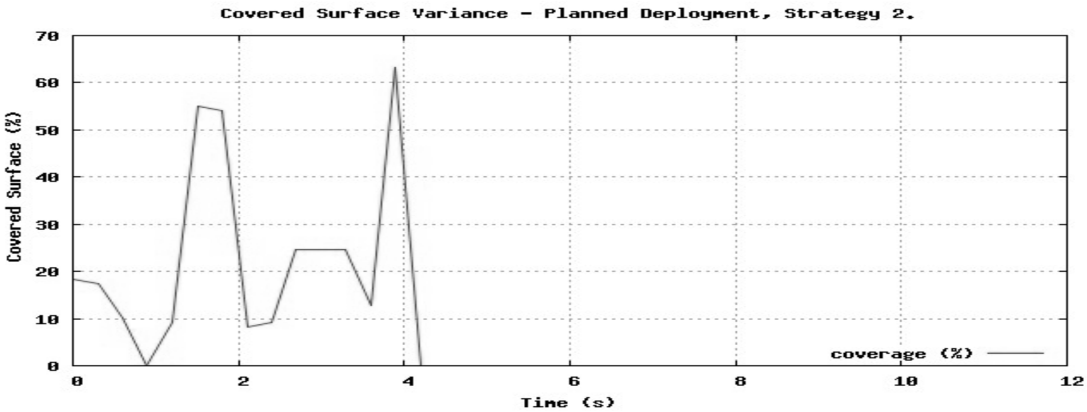
Figure 4.11 represent the case of planned deployment using the Strategy1. Figure 4.12 presents the same strategy, but the sensors I this case are randomly deployed. Figure 4.13 and Figure 4.14 show the test results for Strategy2 in the planned deployment case and in the randomly deployed sensor case. We can observe that there is not a huge difference in terms of coverage between the randomly deployed case and the planned one. The difference is made in a much significant way by the algorithms used.



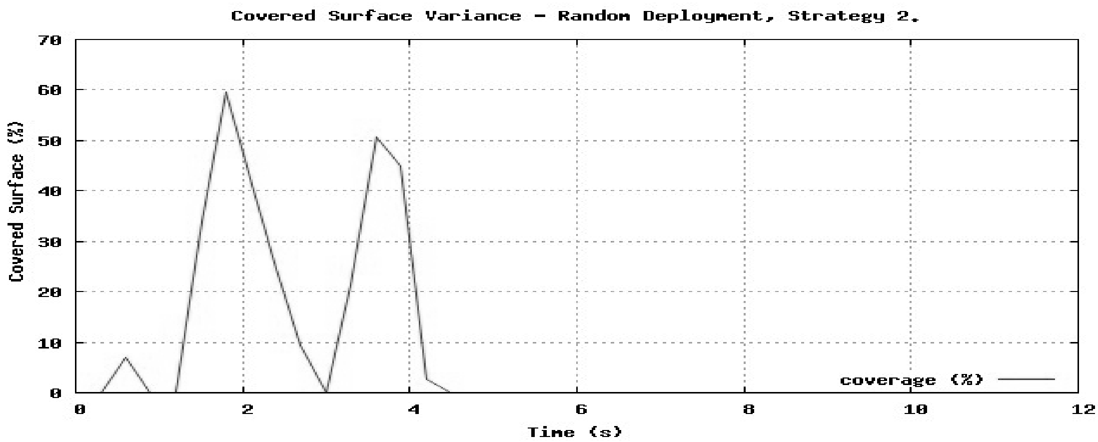
4.11. Planned deployment using Strategy1



4.12. Random deployment using Strategy1



4.14. Planned deployment using Strategy 2



4.14. Random deployment using Strategy2

4.5. Conclusion

This chapter is relevant due to the result analyses that were performed for different wireless video sensor networks scenarios. The difference between random deployment and planned deployment was analyzed and the conclusion reveals the fact that at a sufficient number of deployed sensors (in our case 500) on a limited surface, 1.000x1.000 m2, in the tested scenario, the difference is not relevant.

The number of paths an intruder could take in order to traverse from one edge to another was computed. Solutions to improve coverage degree and realize a good sensor management from the energy perspective was implemented and tested. The efficiency of the proposed method can be seen from the comparisons of figure 4.11 with figure 4.12 and of figure 4.13 to figure 4.14. The proposed method was tested in both planned and random deployment and the results were significant.

Part 2

Performance Efficient Algorithms For Data Collection In WSN

5. Overview on WSN in Traffic Management

5.1. Abstract

The purpose of this chapter is to present the necessity of surveillance in traffic and to show the benefits that are brought by video wireless sensor networks in this direction. Of course there are some requirements and also some limitations. This chapter presents the importance of traffic monitoring and the main approaches in order to obtain efficient traffic surveillance.

In the beginning this chapter presents the need for such an application. Next a number of proposed solutions are presented. The applications of traffic management are classified. The advantages and disadvantages of each approach are presented next.

In addition to the methods specific for each of the domains of the classification, other ways to accomplish traffic surveillance task are discussed.

The drawbacks and limitations are drawn in the conclusion, placed at the end of this chapter.

5.2. Introduction

Real-time data acquisition from broad geographical regions is critical for many applications in transportation, infrastructure management, defense, homeland security, environmental and habitat monitoring, and agriculture [56, 57, 58, 59, 60, 61, 62, 63]. In spite of specific nuances, these systems are similar in that they must collect huge amounts of metadata, e.g., images, sound, temperature, toxin levels, etc., perform local processing, communicate and coordinate with each other through wired and/or wireless networks, and collaborate in achieving global and local goals. Many algorithms in this direction were proposed [64]. Their complex functionality is also subject to stringent performance and design constraints, like hard and soft timing deadlines, sampling and precision requirements, communication bandwidth, and low power and energy consumption [65].

Modern traffic management systems, like adaptive traffic signaling [62, 63, 66], are envisioned to execute complex decision making algorithms that optimize local and global goals, such as maximizing the traffic flow, minimizing the average travel times, minimizing the travel time of "high-priority" vehicles, e.g., fire trucks and police cars, and reducing the vehicle pollution in a zone [67, 68, 69]. Critical to any decision making and control strategy is the acquiring of reliable information in real-time about traffic conditions, like the travel time of vehicles for various road sections, the number of vehicles passing through a zone in a given time, the distances between cars, the vehicle speeds, and the position of traffic incidents [70, 71]. Currently, these parameters are estimated through tedious measurements, and then used in off-line traffic prediction models to compute optimized traffic signal parameters, such as cycle, period, and off-set time [63]. However, current trends in traffic behavior show continuously increasing traffic volumes, more frequent traffic congestions, increased travel delays, high pollution levels, and irregular traffic patterns. These issues are hard to tackle with off-line prediction models and static decision making strategies [62]. Some reports suggest that modern traffic systems present the behavior of complex systems in which new kinds of traffic patterns and behaviors, such as grid locks, can emerge spontaneously. It is increasingly important to develop reliable, online decision making systems that can autonomously operate for a large variety of conditions based on real-time data acquired over broad geographical areas [68, 69, 71].

Based on their sensing devices, traffic monitoring systems can be distinguished into two categories [63]:

(1) Road-based detection systems use sensing devices, like inductive loop detectors and video image detection, and
(2) vehicle-based detection systems require vehicles to be equipped with tracking devices, such as transponders, that allow cars to be tracked by a central server [56].

Video camera based traffic monitoring identifies traffic parameters by sampling and processing images collected through cameras. Video cameras can collect unique information about traffic, such as car color, model, plate numbers, relative position, and passenger occupancy, and in conjunction with other sensors, such as sonars, offer more reliable information about vehicle speeds and positions. Precise estimation of the traffic parameters requires that the monitoring system continuously samples and processes in real-time images that comprehensively describe the ongoing traffic conditions. This is challenging because of the dynamic nature of the process, including vehicles moving at variable speed and obstacles obstructing the field of view of the cameras. Even for slow traffic, image processing and FOV loss recovery must be performed under tight timing constraints (less than one second), if accurate vehicle tracking ought to be secured. Another challenge is due to the unreliable nature of wireless communication, which adds stochastic aspects to the problem. New methods are required to construct solutions with good visual covering under the constraints of dynamic traffic.

5.3. Traffic Management Algorithms

The placement of nodes is important no matter the application they are used for, but in some domains the importance is even higher. One such field is in object tracking or traffic surveillance. In these cases it is really important to cover as much as possible of the target area. Redundant nodes have proven their efficiency not only from coverage perspective, but also from the prolonging lifetime of the network. Moreover there are many applications where, in order to prolong the lifetime of the wireless sensor network, energy saving techniques were used.

Urban traffic management methods are classified into four groups depending on their decision making process:

(1) centralized,

(2) distributed,
(3) hierarchical and centralized, and
(4) hierarchical and distributed [63, 66, 72, 73, 74].

In centralized approaches, decisions are made by a main server. The server acquires input data from all traffic signals, and then computes the control parameters of each traffic signal (e.g., cycle time, split, and offset).

In distributed decision making, all traffic signals are connected in a network, and each decides independently its parameters by monitoring the local traffic and interacting with its neighbors.

In the centralized and hierarchical approach, the central server computes the timing plans for each traffic signal, but each traffic signal can fine tune the plan depending on local conditions.

Finally, the distributed and hierarchical control strategy uses information coming from the central servers as guidance to compute the local control parameters.

Examples of distributed traffic control systems are PRODYN [75], OPAC [76], and SPPORT [77]. The system proposed in [78] is fully distributed in the sense that each local controller receives information only from the local sensors and the neighboring controllers. Thus, only short range communication is required in this approach.

SCATS [79], a hierarchical and centralized traffic control scheme, is arguably the most popular traffic control system. It is currently employed in major cities in USA, Australia, and Europe. SCATS is organized as a two-level hierarchy in which local traffic signal controllers acquire information about the signal flow (using sensors such as inductive loop or video cameras), and send the information over the network to the centralized server. The server computes the minimum, maximum, and optimum cycle times, splits and offsets of each traffic light controller, and then communicates the information back over the network. The server makes decisions using a set of statically defined metrics, like original volume, degree of saturation, and reconstituted volume. A set of rules are used to identify special situations, like congestions. According to [77], SCATS reacts well to short term traffic fluctuations, but is less likely to anticipate future events due to the late acquiring of data on the traffic flow. A similar concept is discussed in [80].

UTOPIA [81] is a hierarchical and distributed traffic management method. More recently, Artificial Intelligence based control methods have been proposed, including methods based on multi-agent systems [69]. The work in [79] suggests a decision making procedure based on propositional calculus and satisfiability (SAT) problem solving. The method encodes the inputs as logic variables, e.g., the traffic volume in different road sections, congestion levels, etc. A set of logic statements define the rules by which the input variables control the decision variables, such as if a traffic signal should change states, or if a two phase or four phase signal cycle should be used. Logic variables are also used for representing qualitative aspects, like traffic levels, congestion, faulty sensing, and idle traffic.

Several methods have been proposed for video camera based traffic monitoring and management. The work in [82] describes Unicam, a video-detection based traffic monitoring system. Two techniques, tripline and tracking, are used for traffic analysis. While the work presents a very interesting solution, the restricted communication layer is one limitation of the method. This is important for monitoring large geographical zones, where a group of cooperating cameras is needed to provide good coverage of the zone.

The approach in [50] focuses on the shadow-induced errors in image acquisition. It detects and suppresses shadows by using the color ratio between the lit and shadowed pixels. A novel traffic congestion monitoring method is proposed in [83].

The method is based on multi agents and pheromone detection. Cars are modeled as agents that deposit pheromone at virtual places. The pheromone evaporates and propagates following a modified version of the state transition model for digital pheromone. Then, a car can predict the traffic on the road ahead from the information provided by the preceding cars.

The system in [56] uses video cameras mounted on buses to dynamically monitor the traffic conditions along traffic corridors. While this work offers interesting solutions, there are still many unanswered challenges, such as the high computational complexity of video processing, and the difficulty of real-time monitoring in the presence of dynamic disturbances (like moving obstacles).

A related topic is video camera based monitoring of geographical areas. Efficient monitoring algorithms have been proposed to maximize the covered area while reducing the used energy [22, 23, 84]. For traffic monitoring, these methods must be changed to address traffic related aspects, such as the impact of the vehicle traffic characteristics on real-time time constraints and data acquisition.

Independent of their data acquiring procedure, effective detection and tracking requires that the application can continuously obtain the current position of any vehicle with a high precision and confidence. This is extremely challenging for modern traffic conditions, where traffic characteristics can change rapidly and many hard-to-predict situations emerge.

5.4. Conclusion

Video sensors are complex devices that perform real time data acquisition. For this reason they have been classified to belong in the field of multimedia. One of the most challenging applications that uses wireless video sensors is traffic surveillance. To achieve the role the sensors are used for, a series of challenges arise. They reflect the need for reliability in the obtained data that was gathered. In this case, reliability means good coverage, real time data processing and transmission and also real time decision making.

These challenges have been addressed in many papers. The solutions can be divided in two main classes: hardware and software. No matter the approach, one aspect that has to be taken into consideration constantly when talking about wireless sensors (especially video sensors) is the amount of limited energy.

The issues that are not yet resolved are the real time data acquisition and the traffic monitoring with smart decision making in order to recover any lost FoV due to obstacles that might appear in front of a sensor. If the FoV is not recovered, coverage drops and significant events might happen and not be seen. Still, despite all these, major advantages in traffic surveillance and intervention are brought by video wireless sensors.

6. Stochastic Model-based Heuristics for Fast Field of View Loss Recovery

6.1. Abstract

In this section we present a novel algorithm for fast FoV loss recovery. The algorithm is analyzed from several perspectives. The first two subchapters present the motivation for this research. The issue debated is presented together with the main concepts used in the proposed algorithm. One key concept is redundancy. The mathematical background for determining the redundancy between sensors is also described.

The FoV loss due to moving obstacles, the performance and resource constraints and the FoV recovery in dynamic conditions are described. The mathematical background for the issues mentioned above are implemented. The issues raised are debated from several points of view. Heuristic algorithms are proposed. One of the algorithms implemented is the algorithm based on linear programming. Linear programming is used to determine the best sensor management and to validate the proposed algorithm in comparison with the mathematical solution. To the presented problems both deterministic algorithm, sustained by the mathematical background and also a stochastic one are presented. Experiments that validate the proposed algorithms are showed and analyzed at the end of this chapter.

6.2. Introduction

This work assumes that there is a set of video cameras available for recovering the FOV loss of a camera due to the moving vehicles. Having a group of cameras monitoring each zone instead of a single camera is a necessary requirement for providing high precision and reliability data acquisition. Our work assumes that one camera could be used in monitoring multiple regions, which reduces the cost of the implementation.

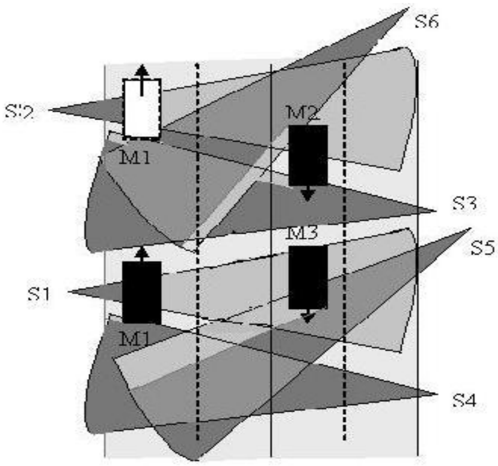
Addressing the FOV loss problem due to interfering objects (e.g., moving vehicles) raises two problems: (a) selecting the subset of additional cameras that must be used to optimally recover the FOV loss, and (b) providing a solution that considers the interdependencies between the FOV losses of neighboring cameras. The two problems must to be addressed in real-time based on on-line video image acquisition on the traffic conditions. The two issues are detailed next.

Selecting the optimal subset of cameras from the total number of cameras that can be used is similar to the knapsack problem, and thus an NP complete problem.

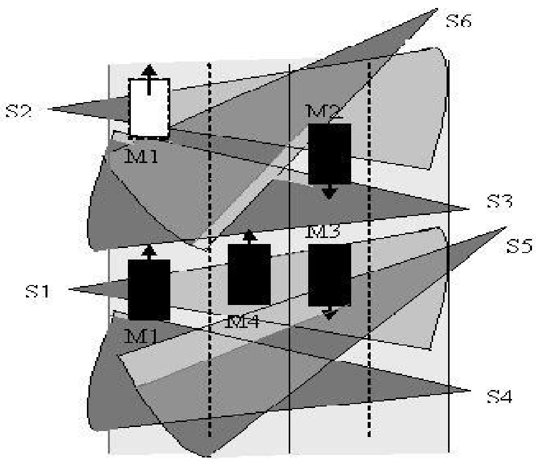
The proposed technique for sensor management was applied for randomly deployed sensors and also along a road for traffic surveillance.

6.3. FoV Loss Recovery in Traffic Management

Selecting the optimal subset of cameras from the total number of cameras that can be used is similar to the knapsack problem, and thus an NP complete problem. Figure 6.1. shows two examples to illustrate the hardness of the problem. Vehicle M1 obscures the FOV of camera S1. The resulting loss is shown using light shading in the figure.



6.1. Coverage loss across multiple cameras



6.2. Coverage loss across for moving vehicles

If other vehicles move across the initial FOV of camera S1, such as vehicles M3 and M4, then these vehicles are not detected by the camera S1. This results in information loss for the entire traffic management system. The FOV loss can be partially recovered by using cameras S3, S4, and S6. However, this set of cameras offers a redundant covering, as the camera pairs S3 and S4, or S4 and S6 are enough in recovering well the FOV loss. Similarly, the coverage loss of camera S2 can be partially recovered through either camera S3 or camera S4 as both can collect the information needed in traffic management. Note that the coverage loss depends dynamically on the traffic characteristics of the moving objects.

The FOV losses of the neighboring cameras are correlated over time. Thus, the camera set assigned for recovering the FOV loss of a camera should consider the FOV loss that occurs for neighboring cameras too. This reduces the related overheads (time and energy) for switching on/off and repositioning the cameras, as the same group of cameras is used to recover multiple FOV losses. This observation is also important for meeting the real-time constraints of video monitoring as it reduces the complexity of FOV loss recovering due to the areas already covered by the cameras identified for the preceding FOV losses. Figure 6.2. illustrates this case. Vehicle M1 obstructs camera S1. Either camera S5 or S6 can recover the FOV loss of camera S1. If vehicle M2 is also moving, as shown in the figure, then it might obstruct camera S5 as it is about to recover the FOV loss of camera S2 due to vehicle M1. However, camera S6 can recover the coverage loss of camera S2. This example explains that the selection method for FOV loss recovering must also consider the FOV losses of the neighboring cameras. These losses occur at future time instances.

6.4. Preliminaries

This chapter presents the characteristics of the context features in which the algorithms are used and tested. They refer to the manner in which the traffic is organized, the number of lanes, the traffic rules.

In order to monitor traffic and develop performant traffic monitoring algorithms, the whole framework has to be as close to reality as possible. To accomplish this, we developed a simulator that is capable of performing traffic simulations with a great diversity of variables that can be set with respect to the traffic conditions that are needed in order to simulate the desired traffic situation.

The idea of simulating with respect to reality or the simulation that tries to reproduce as good as possible the reality is called Monte Carlo simulation. The simulation framework together with Monte Carlo will be described in detail in Chapter 5.

Still, in order to understand the proposed algorithms that are presented next, it is useful to get familiar with the capabilities of the framework, without getting into detail at this point.

The framework offers the possibility of setting the following characteristics regarding the lanes: the number of lanes, the orientation of lanes, the length of lanes, the width of lanes, the direction of cars on each lane, the frequency of the cars on each lane.

The characteristics of vehicles can also be set: the length of the cars, the lane on which the car enters the road, the time to enter the road, the preferred speed for each car and the possibility to choose the type of driving behavior such as speedy driver, slow driver or usual driver.

The simulation of traffic is realized also by setting the minimum distance between cars and the overtaking distances at which the overtaking is allowed.

All these aspects are implemented. In order to obtain a realistic simulation, we used real data. We used the real data also for setting the above mentioned variables.

Sensors are used to monitor traffic along the road and can be considered scattered or placed at will. The sensors collect data regarding the traffic and send it to the central unit.

6.5. Redundancy

The motivation to develop this algorithm came from the fact that in a WSN, the number of scattered nodes is huge, approximately up to 20 nodes/m3 [86].

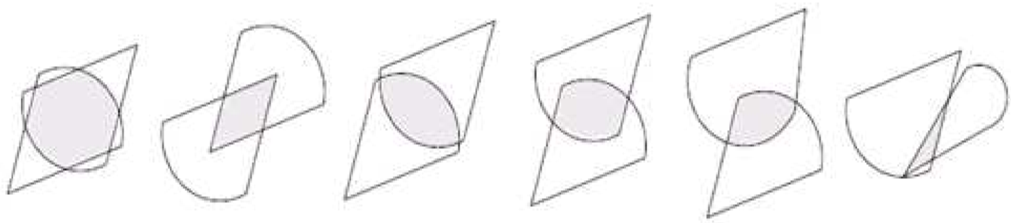
In the proposed algorithm, we considered a node to be redundant only if the area that was covered by that node is covered by at least one more node in a percentage greater than 70%. The nodes being scattered, the deployment cannot be controlled with grate precision, so it is lightly that for a significant number of nodes, some areas to be covered by more than only one or two fields of view of the sensors. In this case, the algorithm detects the intersection of all the sensors and decides which sensor is the most insignificant. Only that sensor is turned off and this happens only in the case that the overlapping area for that specific sensor is greater than the specified percentage.

Although the redundant nodes are turned off for a period of time, the total area coverage is checked before each turning off of the nodes. This way, we ensure that we save energy, but we still maintain coverage within desired limits.

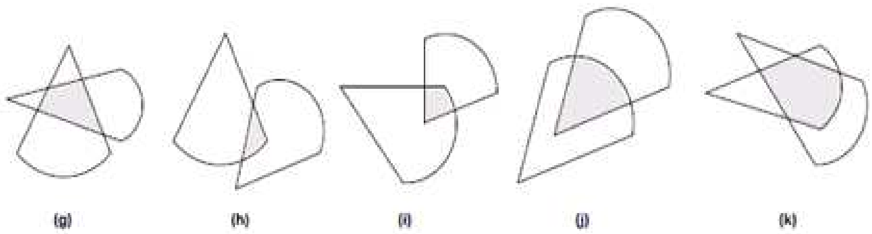
The problem of determining the intersection for the fields of view of two sensors was resolved mathematically. Each sensor was considered to be a sector of a circle. This way, the problem became finding the area of the intersection for two sectors of circles.

Using AutoCad [87] we have determined that there are 11 main possibilities for the intersection of two sectors of a circle. These cases are general. The particular cases were considered to be resolved automatically and were considered to belong to the general case.

The cases found are presented below in Figure 8.3.

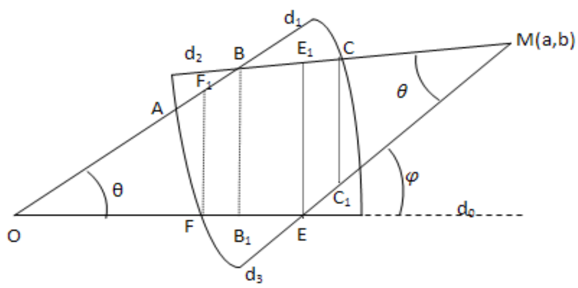


(a) (b) (c) (d) (e) (f)



6.3. Possible FOV intersections for two cameras

6.5.1. Mathematical background



8.4. Detailed description of the intersection cases (a)

The intersection in Figure 8.4. (a) the area is as follows,

$$A_{AFF1}+A_{FF1BB1}+A_{BB1EE1}+A_{E1EC1C}+A_{C1CD} \tag{1}$$

The equations of the lines are:

$$d_0: y=0 \tag{2}$$

$$d_1: y=x\,tg\theta \tag{3}$$

$$d_2: (y-b)=(x-a)tg(\varphi-\theta) \tag{4}$$

$$d_3: (y-b)=(x-a)tg\varphi \tag{5}$$

Finally, the two circles are described by the following two equations:

$$C_1: x^2+y^2=R^2 \tag{6}$$

and

$$C_2: (x-a)^2+(y-b)^2=R^2 \quad (7)$$

Then, the five areas in formula (1) are given by the following expressions:

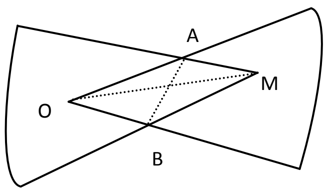
$$A_{AFF1} = \int_{x_A}^{x_F} xtg\theta \, dx - \int_{x_A}^{x_F} (\sqrt{R^2-(x-a)^2}+b) \, dx \quad (8)$$

$$A_{FF1BB1} = \int_{x_F}^{x^B} xtg\theta \, dx \quad (9)$$

$$A_{BB1EE1} = \int_{x_B}^{x_E} xtg(\varphi-\theta) \, dx \quad (10)$$

$$A_{E1EC1C} = \int_{x_E}^{x_C} xtg(\varphi-\theta) \, dx - \int_{x_E}^{x_C} xtg(\varphi) \, dx \quad (11)$$

$$A_{C1CD} = \int_{x_C}^{x^D} (\sqrt{R^2-x^2}) \, dx - \int_{x_C}^{x_D} xtg\varphi \, dx \quad (12)$$



6.4. Detailed description of the intersection cases (b)

For the case in Figure 6.4(b), the area is computed by the following expression

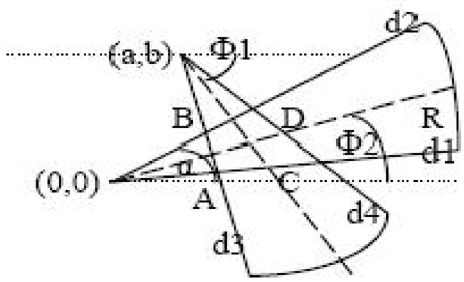
$$\frac{OM \;\; * \; AB \;\; * \sin(\; OM \;\; * \; AB \;)}{2} \quad (14)$$

where

$$OM = \sqrt{x_M^2 + y_M^2} \quad (15)$$

$$AB = \sqrt{(x_A-x_B)^2+(y_A-y_B)^2} \quad (16)$$

Points A and B are computed similar to the previous case, based on the intersection points between lines.



6.4. Detailed description of the intersection cases (c)

The mathematical formulation of the intersection in Figure 6.4. (c) is as follows. For each sensor, the known parameters include the angle α (the opening angle of the camera), the angle φ (the angle of the sensor with respect to the axis x), the radius R of the sensor, and the position of the sensor. The intersection points of the two sensors were computed using the angular coefficient. For example, point A is the intersection of lines d1 and d3. The equation of line d1 is given next:

$$y = mx \tag{17}$$

where

$$m = tg\left(\Phi 2 - \frac{\alpha}{2}\right) \tag{18}$$

The equation for line d3 is expressed as follows:

$$y - b = p(x - a) \tag{19}$$

where

$$p = tg\left(180 - \left(\Phi 1 + \frac{\alpha}{2}\right)\right) \tag{20}$$

The position of the intersection point A results from equating equations (17) and (19):

$$x = \frac{pa + b}{m - p} \text{ and } y = m\,\frac{pa + b}{m - b} \tag{21}$$

The other intersection points were computed similarly.

The intersection area of the two disk sectors is as follows:

$$A = \int_{\frac{b-m'a}{m-m'}}^{\frac{b-pa}{m-p}} mxdx - \int_{\frac{b-m'a}{m-m'}}^{\frac{m'a-b}{m'}} (b + m'x - m'a)dx + \int_{\frac{b-pa}{m-p}}^{\frac{pa-b}{p}} (b + px - pa)dx, \quad (22)$$

where

$$m' = tg(\Phi_2 + \frac{\alpha}{2}) \quad (23)$$

Finally, for case (k), the intersection area is expressed by the next formula:

$$A = \frac{\alpha^2 R^2}{2} - S \quad (24)$$

where

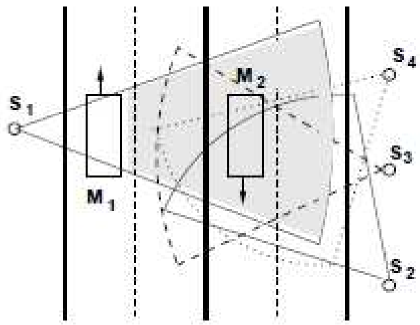
$$S = \int_{\frac{b-m'a}{m-m'}}^{R\cos\alpha} [mx - b - m'(x-a)] \cdot dx + \int_{R\cos\alpha}^{x_1} \sqrt{R^2 - x^2}\, dx \quad (25)$$

$$\int_{R\cos\alpha}^{x_1} \sqrt{R^2 - x^2}\, dx = \frac{R^2}{2} \arcsin\frac{x}{R} + \frac{x}{2}\sqrt{R^2 - x^2} + C \quad (26)$$

and $\Phi 2 - \frac{\alpha}{2} = 0$.

Similar expressions were derived for the other intersections.

6.6. Video Camera Based Traffic Monitoring Model



6.5. FOV Coverage Loss Across Multiple Cameras

Cameras lose their FOV due to moving obstacles. For example, in Figure 6.5., vehicle M1 obscures the FOV of camera S1, and hence vehicle M2 cannot be tracked. The resulting FOV loss is shown with light shading in the figure. Note that the coverage loss depends dynamically on the tracked vehicles, their position with respect to the camera, the speed of the vehicles, and number and traffic characteristics of the moving obstacles.

The FOV loss in the figure can be partially recovered by using cameras in the set S1,F = {S2, S3, S4}. Recovering the FOV loss of a group of cameras requires finding dynamically the smallest set of additional cameras that offer the best loss recovery while meeting all timing constraints imposed by the sampling requirements of the application, and minimizing the used resources, such as communication bandwidth and energy. Specifically, for any camera Si of a monitored region, the set Si,F of cameras used in FOV recovery must be dynamically computed for every time instance t, so that the total remaining FOV loss (the loss not recovered through set Si,F) is minimized:

$$\sum_{S_i} \int_{\infty} \left(FOV\ Loss_i(t) - \sum_{S_j \epsilon S_{S_i,F}} FOV\ recoveryS_j(t) \right)^2 dt \qquad (27)$$

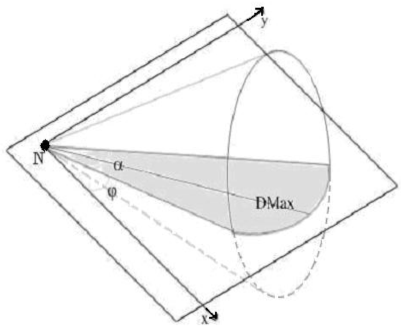
Set V is the set of all cameras of the monitored region. The minimization objective in expression (1) gives more weight to cameras with large FOV losses. Alternatively, the following objective function treats equally every camera to give a uniform relative recovery to each camera $S_i \in V$:

$$\sum_{S_i} \int_{\infty} \frac{1}{FOV\ Loss_{S_i,total}(t)} \sum_{S_j \epsilon S_{S_i,F}} FOV\ recoveryS_j(t)\ dt \qquad (28)$$

This problem must be solved under dynamic conditions due to the moving of obstacles in time and under performance and resource constraints. Solving the problem in static conditions is similar to the minimum set covering problem, which is known to be NP complete [22, 23, 84]. This section presents the model used for expressing the dynamic FOV loss recovery problem. Next section uses the model for defining the proposed heuristic algorithms.

6.6.1. Background on FOV Loss Due to Moving Obstacles

We assume that the cameras in set V are placed on the same horizontal plane. Consequently, all FOVs are 2D projections of the 3D volume of the view on the planar surface. All points in front of the camera are visible as long as the projection ray from the point to the optical center intersects the image. In practice, however, due to the limited resolution and distortion of their lenses, cameras have a bounded depth of field. If the points that are too close or too far from the optical center may not appear well focused. In order to get an accurate perspective, the used model considers that an object is in the FOV of a camera, if the object is within the range (DMin,DMax), where distances DMin and DMax depend on the video sensor resolution and the minimum size of the target of interest. Figure 6.6. illustrates the FOV model for a camera.



6.6. Field of view (FOV) model

By processing its sampled image, any camera Si detects the presence of moving obstacles (vehicles), and thus determines its FOV loss. The amount and nature of the loss is estimated based on the size and position of the obstacle. The FOV loss for a certain vehicle type M is function FOV LossM(position, t), where position is the 2D position of a vehicle relative to the camera at time t:

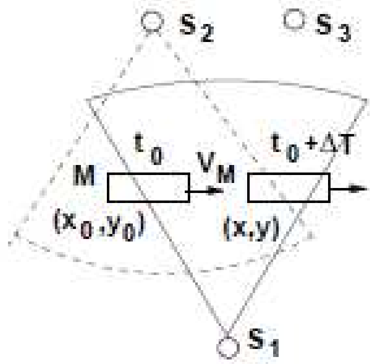
$$FOV\ Loss_M(x, y, t): R_+^3 \rightarrow Arca \tag{29}$$

x and *y* are the coordinates of the obstruction and t is the current time. Area is the lost FOV area due to the obstruction. The values of the function can be pre-computed for all possible vehicle types and their positions. To save camera resources, the proposed method considers that a camera stops monitoring, if its FOV loss is more than k% of its total FOV. In such as case, the camera is turned off until its FOV rises again above k%.

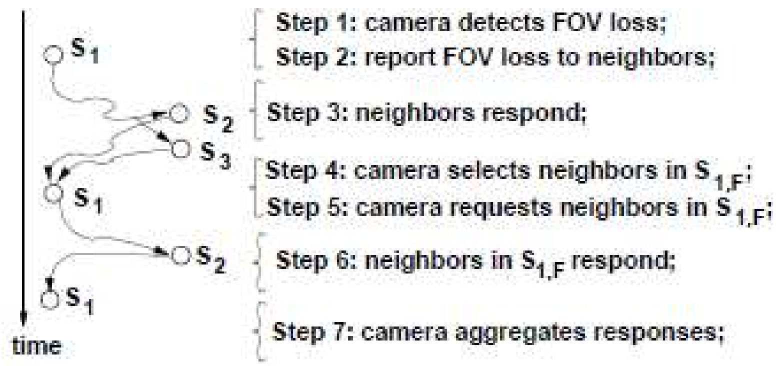
Even though a camera might be off, its communication subsystem is constantly on to forward and receives any request to power on a stopped camera. For a camera pair, the mathematical expressions of the FOV loss recoveries can be pre-computed for every position of the pair, and then used in selecting the cameras to be turned on to recover a FOV loss.

6.6.2. FOV recovery under performance and resource constraints

Video cameras communicate with each other by sending information about their FOV loss, and then collaborate to recover the experienced FOV loss. The recovery time until the additional cameras in set $S_{i,F}$ can compensate the FOV loss depends on the execution time of the image processing algorithms performed by the cameras, the communication delay between cameras, the time needed to switch on any cameras turned off, and the time required to reposition the cameras.



6.7. Vehicle Dynamics



6.8. Interaction Scheme

Figure 6.8. summarizes the proposed interaction scheme between cameras. The scheme shows the sequence of steps over time, and the actions performed by the cameras. For example, if camera S1 in Figure 6.7. loses part of its FOV due to vehicle M then the loss can be recovered by cameras S2 and S3 (S = {S2, S3}). The interaction procedure between the collaborating cameras is as follows. Camera S1 informs the neighboring cameras, e.g., cameras S2 and S3, about its FOV loss. Steps one and two in Figure 8.8. require a constant execution time $T_{S_1}^{1,2}$, which includes the processing time $T_{S_1}^{reg}$ of camera S1 to detect its loss, and time $T_{S_1}^{2}$ for broadcasting the information. Then, the neighboring cameras respond by indicating what area of the loss they can recover. The fourth step consists of camera S1 receiving the information from neighbors and deciding which of them to select for the FOV loss recovery. The cumulative execution time of steps three and four is equal to

$$T_{S_1}^{3,4} = max_{i\in\text{neighbors}(s_1)} T_i + \sum_{i\in neighbors(S_1)} Decision_{S_1}(i)$$

The first terms is the time required by S_1's neighbors to respond, and the second term is the time needed by camera S_1 to analyze the responses and make a decision. In step five, camera S_1 requests a smaller subset $S_{1,F}$ of cameras to stream their actual images ($S_{1,F} = \{S_2\}$ in the figure). Then, in step six, the requested cameras stream the image fragments, which are aggregated by camera S_1 in step seven. The execution time of the last three steps is

$$T_{S_1}^{5,6,7} = T_{S_1}^{reg} + \max_{i\in S_{1,F}} T_i^{\text{stream}} + \sum_{i\in S_{1,F}} \text{Agregate(i)}$$

The second term is the time for streaming the images for loss recover, and the third term indicates the time for aggregating the received images.

If images must be collected with a period $T_{constraint}$ then all steps related to the current FOV loss of a camera i must be completed before the next image sampling:

$$T_{S_i}^{1,2} + T_{S_i}^{3,4} + T_{S_i}^{5,6,7} \leq T_{constraint} \quad (30)$$

In addition, to provide meaningful aggregation, the images sampled by the neighbors must be within a time window from the current moment:

$$T_{S_1}^{1,2} + \max_{i\in S_{1,F}} T_i \leq T_{aggreg} \quad (31)$$

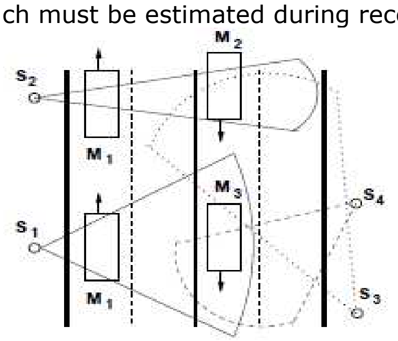
In addition to constraints (30) and (31), the loss recovery is also subject to timing constraints that are correlated to the speed of the moving vehicles. For example, if camera S_1 in Figure 6.6 must collect n images while vehicle M traverses its FOV then the worst-case timing constraint for sampling and processing one image is expressed by the following formula:

$$T_{constraint} \leq \frac{d}{v_{M,max} n} \quad (32)$$

where parameter d is the length of the vehicle's trajectory within the camera's FOV, and value $v_{M,max}$ is the maximum speed of the vehicle. If the number of samples n is not a hard constraint then the relation can be relaxed to include the expected speed of the vehicle $E[v_M]$:

$$T_{constraint} \leq \frac{d}{E[v_M] n} \quad (33)$$

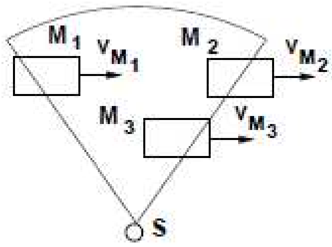
Note that a larger sets S and $S_{i,F}$ improve the quality of the possible loss recovery, however they increase the time and resource overhead of the method due to the more alternatives that have to be analyzed. The FOV loss recovery method must identify which camera subset $S_{i,F}$ of set S offers the best recovery within the set time constraint and resources, e.g., processing speed and communication bandwidth. A second issue is related to the reliability of wireless communication. Wireless links tend to be unreliable, and therefore the availability of the cameras in set $S_{i,F}$ can change over time. The decision about which cameras to use for FOV loss recovery must also consider the likelihood of a camera to be available or not in the immediate future. Besides, the arrival time of the camera responses might become random due to changing communication conditions. Some responses might be discarded because they arrive after the imposed timing deadlines. Thus, set $S_{i,F}$ has a stochastic character, which must be estimated during recovery.



6.9. FOV Recovery Loss Over Time

6.6.3. FOV recovery in dynamic conditions

The FOV loss of neighboring cameras is correlated in time by the specific traffic conditions, such as the number and speed of moving cars. The information about the current traffic conditions can be used as look ahead information to estimate the position of the moving obstacles at future instances of time. Estimations can be used to decide the assigning of the additional cameras to the best set $S_{i,F}$, if a camera can be participate to the FOV loss recovery of multiple cameras. The information about the expected FOV loss at neighboring cameras and at future instances of time can also help reducing the execution time of loss recovery by limiting the candidate set S to cameras, which are estimated to offer the highest recovery. Figure 6.9 illustrates this situation. Vehicle M_1 obstructs camera S_1 at the current moment. Either camera S_3 or S_4 can recover the loss of camera S_1. If vehicle M_1 is moving as shown in the figure, then at a future moment it also obstructs camera S_2. Note that camera S_3 can recover the coverage losses of both camera S_1 and S_2. Camera S_3 is included into set $S_{1,F}$ at the current moment and set $S_{2,F}$ at the next moment. Camera S_4 becomes available for recovering the loss of other cameras. The vehicle dynamics (trajectory and speed) also determines the FOV loss of a camera over time as long as the vehicle stays in the camera's FOV. Figure 6.7. presents the situation in which the loss at time instance $t_0+\Delta T$ depends on the position of vehicle M at time t_0.



6.10. Traffic Modeling1

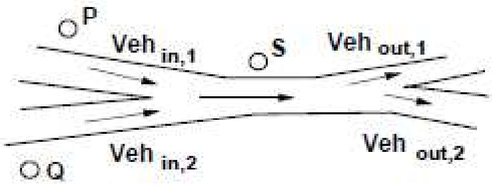
The FOV loss recovering model must capture not only the losses of a camera at the current moment but also the expected FOV losses of the neighboring cameras at later time instances. The model expresses the expected FOV loss at the next time moment t_0+ΔT depending on the loss at the current moment t_0. Figure 6.10. shows the modeling of the cumulative FOV loss of camera S due to multiple vehicles. Each of the three vehicles causes FOV loss. The overall FOV loss of the camera is as follows:

$$FOV\ Loss_S(t_0 + \Delta T) = \sum_{i \in Veh(t_0+\Delta T)} FOV\ Loss\ by\ Vehicle_{M_1}\big(position\big(M_{i,}t_0 + \Delta T\big), t\big) \quad (34)$$

where set *Veh* is the set of the vehicles M_i crossing the camera's FOV at time *t*. The set *Veh* of vehicles obstructing camera *S* at the future moment t_0 +ΔT depends on the vehicles that enter the camera's FOV and those that exit. Figure 6.11. shows the regions monitored by cameras *P, Q*, and *S*. The cars entering region *S* are the cars leaving regions *P* and *Q*. Then, set *Veh* at time moment t_0 +ΔT is expressed as follows:

$$Veh(t_0 + \Delta T) = Veh(t_0) \bigcup \left(\left(\bigcup_{t_0, t_0+\Delta T} Veh_{in} \right) - \left(\bigcup_{t_0, t_0+\Delta T} Veh_{out} \right) \right) \quad (35)$$

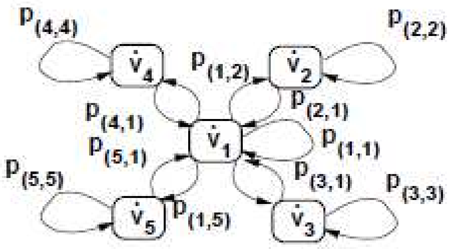
where V*ehin* are the vehicles entering into the zone, and Veh_{out} are the vehicles leaving the camera's FOV.



8.11. Traffic Modeling2

The position of vehicle M must be estimated in order to predict the future FOV loss of the camera. The future position of the car depends on the traffic conditions and driver's behavior. Without reducing generality, let's assume first that the car's movement is not constrained by the moving of other cars. This is the case of light traffic. Then, the future position depends mainly on the driver's behavior to change the speed during the time range (t_0, $t_0+\Delta T$). We model the driver's behavior as a Continuous Time Markov Process (CTMP), as shown in Figure 6.12.:

$$CTMC = (R, A, A(i), p, K, r) \tag{36}$$



6.12. Vehicle dynamics modeling

where *R* is the set of states, which correspond to the highly probable speed gradients v_i. Gradient $v_1 = 0$ describes the moving with constant speed. *A* is the action set, and *A(i)* are the actions associated to state $i \in R$. The action sets are empty in this model. *p(i, j)* is the transition rate for transitioning from state *i* to state *j.* The transition probabilities between states describe a certain driving profile, and are pre-computed. CTMP *K* is the number of reward criteria, and $r_k(i)$ is the reward rate for state *i*. These sets are empty in our model. x_i is the steady-state probability of state *i*. The following set of equations describes the driver behavior:

$$p(i,i)x_i - \sum_{j \in R} p(i,j)x_i = 0, \forall j \in R, \tag{37}$$

$$\sum_{i \in R} x_i = 1, \forall i \in R \tag{38}$$

$$x_i \geq 0, \forall i \in R \tag{39}$$

Unknowns are the steady-state probabilities x_i.

The expected vehicle speed at time $t_0 + \Delta T$ is modeled by the expression:

$$E[v(t_0 + \Delta T)] = v(t_0) + \Delta T \sum_{i \in R} v_i x_i \tag{40}$$

Hence, the expected position of vehicleM at time instance $t_0+\Delta T$ is as follows:

$$E[position(t_0 + \Delta T)] = position(t_0) + \Delta T\ E[v(t_0 + \Delta t)] = position(t_0) + \Delta T\ v(t_0) + \Delta T^2 \sum_{i \in R} v_i x_i \tag{41}$$

The expected position is replaced in function FOV Loss in equation (34) to compute the expected FOV loss at time t_0 +ΔT. The vehicles' traffic parameters, such as speed, might be correlated depending on the specific traffic conditions. If the traffic is light then each vehicle can move without being constrained by the other vehicles. Hence, the transition probabilities in equation (37) are independent, and depend only on the driver's behavior. In contrast, for heavy traffic, cars are clustered together, and then the transition probabilities of all cars in a cluster tend to be the same as those of the car in front as the cluster moves with the same speed. The interaction between the vehicles in regions *P, Q*, and *S* in Figure 6.11. can be expressed depending on the cardinality of sets Veh_{in}. If $\sum_i Cardinality(Veh_{in,i})$ is less than a threshold value then the traffic is considered light, and the dynamics of each vehicle can be estimated individually. Otherwise, the traffic is heavy.

6.7. Heuristic Algorithms for FOV Loss Recovery

The FOV loss recovery problem described as a discrete optimization problem with constraints. Moreover, the nature of wireless communication and the dynamics of moving obstacles introduce stochastic elements. Without expressing the stochastic aspects, the problem is similar to constrained minimum set covering, which is known to be NP-complete [21, 22, 23]. This section presents three approaches to solving the problem. The first method is based on ILP (Integer Linear Programming) formulation of the problem, and provides the starting formulation of the problem. However, it considers only a simplified scenario. The second method is an heuristic algorithm that tackles performance and resource constraints and the dynamics of the vehicles. Finally, the third heuristic method also assumes the stochastic nature of wireless communication. The cost functions in formulas (27) and (28) are approximated by considering only the current moment t_0 and the next time instance t_0+ΔT. For every camera S_i, all timing constraints related to image sampling must be satisfied, e.g., equations (30), (31), (32), and (33). In addition, the methods minimize the number of candidates for FOV loss recovery (in set $S_{i,F}$). Reducing the number of candidates indirectly helps meeting the timing constraints, and also lowers the amount of used resources, such as communication bandwidth and energy consumption. Finally, the methods lower the involved overhead by selecting candidates that are useful over longer periods of time and for multiple obstructed cameras. This requires predicting the set V eh of obstructing vehicles (using expression (35)) and computing the expected positions of the moving vehicles at future instances of time (with expressions (41) and (40)). The parameters expressing the dynamics of different vehicles can be uncorrelated or correlated depending on the specific traffic conditions, such as light or heavy traffic. Note that a central issue of the problem is the management of sets $S_{i,F}$ used to recover the FOV loss of camera S_i. Cameras in set $S_{i,F}$ must be selected to provide the best recovery under time constraints and while reducing the utilized resources, like bandwidth, and considering the stochastic nature of wireless communication. The allocation of cameras to alternative $S_{i,F}$ depends on the dynamics of the FOV loss over time.

6.7.1. ILP-based algorithms

Arguably, the most intuitive approach is to formulate the optimization problem as an ILP equation set [22, 23]. We use this method as a reference. An ILP formulation is possible if the intersections between camera pairs are statically defined so that they can be pre-calculated (coefficients $a_{i,j}$ in expression (18)). For a given loss, the objective is to maximize the expression:

$$\boldsymbol{max}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} \boldsymbol{x_i}\,(\sum_{\boldsymbol{j=1}}^{\boldsymbol{p_1}} \boldsymbol{a_{i,j}}) \tag{42}$$

where N is the number of cameras. Coefficients $a_{i,j}$ are the areas of the p_i sub-regions of the FOV losses that are covered by camera S_i. The FOV loss includes the loss at time t_0 and time $t_0+\Delta T$. Variables x_i are 0/1 variables that have value zero if camera S_i is not used, and value one, otherwise. By solving the ILP equations, the algorithm finds the values of the unknowns x_i (hence, if camera S_i is on or off) that maximize the objective. In addition, the following constraints must be satisfied for every sub-region j of the loss:

$$\sum_{i \in G_j} x_i \leq \gamma \tag{43}$$

where G_j is the set of cameras that can cover the sub-region j, and constant is the maximum amount of redundant covering that is accepted due to the need to save resources (was set to value one in our experiments).

If the intersections between cameras change dynamically, such as due to mobile cameras and adding/removing cameras, then the problem becomes nonlinear as parameters $a_{i,j}$ are also unknown. The stochastic nature of wireless communication is also not captured in the basic formulation.

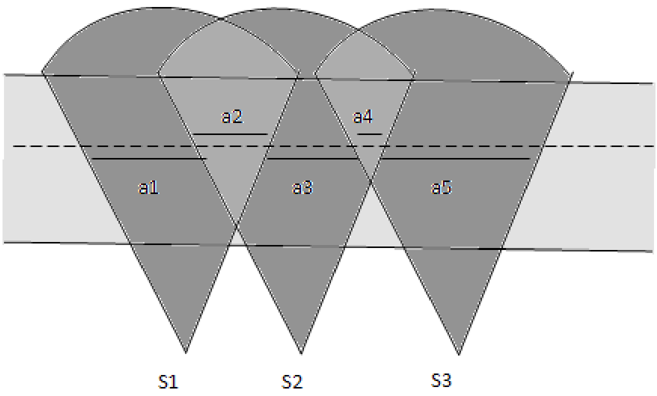
ILP refers to linear programming. We used linear programming to optimize the selection of the sensor that would be turned on in order to replace the obstructed one.

The implementation of ILP concepts was realized with LPSolve [22, 23]. For this equation corresponding with Figure 8.13., the coverage is given by the following formula:

$$a = a1\ S1 + a2\ (S1 + S2) + a3\ S2 + a4\ (S2 + S3) + a5\ S3 \tag{44}$$

The equation to be maximized is max(a). The constraints for these equations are:

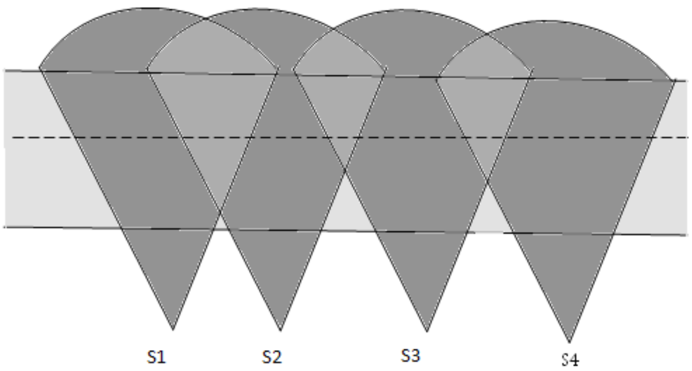
$$\begin{cases} S1+S2=1 \\ S2+S3=1 \end{cases} \Rightarrow \begin{cases} S1=0 \\ S2=1 \\ S3=0 \end{cases} \text{ or } \begin{cases} S1=1 \\ S2=0 \\ S3=1 \end{cases} \tag{45}$$



6.13. General case study for 3 sensors

The last set of equations represents the maximized solution because two sensors are on, instead of one.

The second case considered is the situation where we have more sensors. This is important due to the fact that more redundancy groups are formed. We are interested, in this case, as well, to maximize the coverage equation.



8.14. General case study for more than 3 sensors

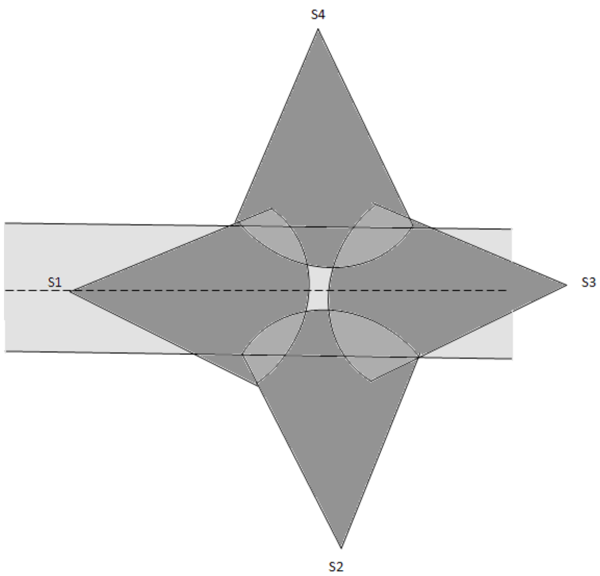
For Figure 6.14. the constraint equations will be in this case constraint inequations:

$$\begin{cases} S1+S2 \leq 1 \\ S2+S3 \leq 1 \\ S3+S4 \leq 1 \end{cases} \Rightarrow \begin{cases} S1=1 \\ S3=1 \\ S2=0 \\ S4=0 \end{cases} \text{or} \begin{cases} S2=1 \\ S4=1 \\ S1=0 \\ S3=0 \end{cases} \text{or} \begin{cases} S1=1 \\ S4=1 \\ S2=0 \\ S3=0 \end{cases} \text{or} \quad S1,S2,S3,S4 \quad (46)$$

In this case the algorithm computes which of the above solutions has the best coverage and chooses one of the first three solutions. The last one is not considered because we are interested in the greatest number of sensors that are turned on that have no redundancy.

Another example would be described by the following case where the sensors are not aligned – Figure 8.15. For this case the constraint equations are:

$$\begin{cases} S1+S2=1 \\ S1+S4=1 \\ S2+S3=1 \\ S3+S4=1 \end{cases} => \begin{cases} S1=1 \\ S3=1 \\ S2=0 \\ S4=0 \end{cases} \quad \text{or} \quad \begin{cases} S2=1 \\ S4=1 \\ S1=0 \\ S3=0 \end{cases} \tag{47}$$



6.15. General case study for more than 3 sensors not placed in line

In the performed study, we implemented and used Lp Solve to determine the best case of sensor selection by maximizing the coverage equation. The solution provided by this method is the best from the coverage perspective, but it was used only to prove the performance of the proposed algorithms by comparison due to its major drawback. The main disadvantage of ILP implementation in practice is the fact that the computation of all the equations is time demanding and need performant processors. Unfortunately this would not be suitable to implement on real wireless sensors. Moreover, due to the time needed for computation, the latency would not respect the real-time constraint anymore and by the time the computation of which sensor is most suitable to be turned on in order to recover the FoV of the obstructed sensor, the car might already be out of the area of interest. Still, as mentioned above, the method is used for comparing its results that are mathematically the best decisions with the proposed algorithms and prove the performance of these algorithms.

The next two heuristic algorithms adapt greedy, minimum set covering strategies to our problem. A good cost function for minimum set covering is to always select the candidate that covers most of the uncovered elements [22]. The main advantage of heuristics is short execution time, which is important for meeting real-time constraints. The two algorithms differ by their cost functions used in greedy selection. The two cost functions are derived from the model presented above for two different scenarios, deterministic and stochastic scenarios.

6.7.2. Heuristic algorithm 1 (deterministic)

The first cost function assumes that the cameras are interconnected through a fast and reliable network (e.g., wired connections). Thus, it is reasonable to assume that all candidates can submit their data within the real-time constraint and without experiencing any data loss. The proposed cost function, called Parameter Weighted Contribution (PWC), characterizes the suitability of a neighboring camera *X* to cover the FOV loss of camera *Y* . Based on the model presented above, PWC captures the following aspects: (i) camera *Y* 's FOV loss ratio that is covered by camera *X*,

(ii) the uniqueness of the FOV coverage provided by camera *X* compared to the other cameras in set *S*,
(iii) the capability of camera *X* to cover the FOV loss of other cameras than camera *Y* (including any FOV loss due to obstacle
dynamics in the near future), and
(iv) the available resources of camera *X* (e.g., energy, communication bandwidth, and processing speed). PWC is defined as follows:

$$PWC = \delta COV + \gamma UCOV + vFCOV + kRES \tag{48}$$

Where δ, γ, v and k are weights. The other parameters of PWC are defined as follows:

- Parameter COV is the ratio of the FOV area covered by camera *X* over the area of the FOV loss of camera *Y* :

$$COV = \frac{FOV\ covered\ by\ camera\ X}{FOV\ loss\ of\ camera\ Y} \tag{49}$$

Parameter *COV* maximizes the objective function in equation (27). Maximizing this parameter also minimizes the cardinality of the covering set, thus reduces time $T^{5,6,7}$ in the presented model. This helps satisfying the timing requirement in expression (30). It also reduces the communication bandwidth because less cameras participate to the loss recovery. This parameter is similar to traditional priorities in minimum set covering heuristics [22], which choose subsets with many uncovered elements.

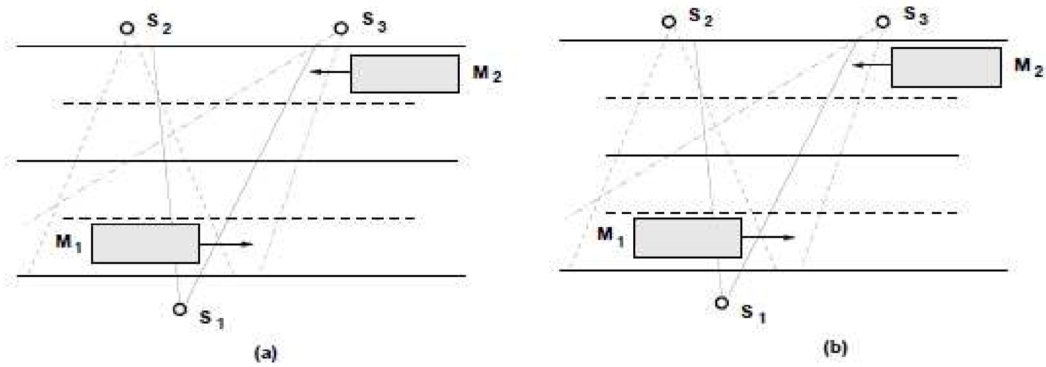
- Parameter UCOV characterizes the uniqueness of camera *X* to cover the FOV loss of camera *Y* :

$$UCOV = \frac{Area\ covered\ by\ camera\ X}{Number\ of\ other\ cameras\ that\ cover\ the\ same\ FOV\ loss\ as\ camera\ X} \quad (50)$$

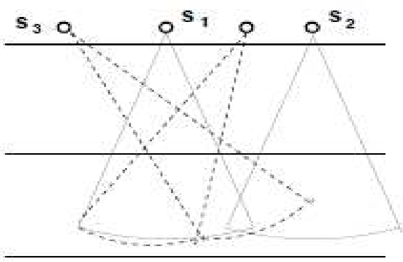
If a sensor is unique in covering a certain loss than that sensor is selected with a high priority. This parameter decides the inclusion of a camera to a certain set $S_{i,F}$, if the camera can be used to recover the FOV loss of multiple cameras.

- Parameter *FCOV* captures the usefulness of the sensor *X* in predictably covering future FOV losses of neighboring sensors (i.e. due to moving vehicles). The parameter refers to situations for which reliable predictions can be made about neighboring cameras losing their FOV too. More details are offered in the next paragraph.

- Parameter *RES* captures the resources available at sensor *X*, like the processing speed, communication bandwidth, and energy. The algorithm tries to avoid using cameras that are slow, low on energy, or have low communication bandwidth (e.g., weak signal strength). These cameras are used with a low priority, such as special situations like if the camera is the only one that can observe a given area. The parameter also captures the capability of a candidate camera to sample an image within the timing constraint expressed by equation (31) and to provide sufficiently fast sampling as described by equations (32) and (33).



6.16. Definition of priority functions



(c)

6.17. Definition of priority functions

Parameter FCOV is detailed next. Our discussion refers to two illustrating cases on preserving the FOV coverage during a certain time interval. The first case considers three sensors S_1, S_2 and S_3 that cover the same FOV, and two vehicles move in opposite directions, as shown in Figure 6.16.(a). Camera S_1 is covered by vehicle M_1, and thus has to decide whether to start camera S_2 or camera S_3. Without estimating any future situations, camera S_3 is started as its parameter PWC is large. However, camera S_3 loses its coverage in a short time since vehicle M_2 is moving in its direction. Hence, parameter PWC should be defined so that camera S_2 is selected in this case. In the second case, the moving vehicles are in the same lane, as shown in Figure 6.17.(b). Switching between cameras S_1 and S_2 is a good solution, if only vehicle M_1 is considered. If vehicle M_2 is also considered, then selecting camera S_3 to be turned on is a better solution over long time periods.

The definition of parameter FCOV covers the previously described requirements. The parameter increases with the estimated time of loosing coverage in the future because of close vehicles. The estimated time is proportional to the ratio of the distance d between the camera and the obstacle over the estimated speed E[v]:

$$FCOV = \sum_{neighboring\ sensors} (COV + UCOV) - \frac{d}{E[v]} \quad (51)$$

The speed is estimated as described in equations (36)-(39).

6.7.3. Heuristic algorithm 2 (stochastic)

The second heuristics assumes slow and unreliable connection between the video cameras, such as in the case of wireless communication. This implies that the communication time represents a significant portion of the left-hand side in equation (30). Moreover, the arrival times of the responses from a camera's neighbors are random, including situations in which a response might not be received on time due to poor communication conditions. Hence, the set *neighbors(S_i)* of any camera S_i is not fully deterministic as some of the neighbors might "disappear" for certain time periods.

The heuristic algorithm analyzes the currently received data from the neighboring cameras, and selects the cameras that are more likely to participate to the FOV loss recovery. In contrast to the deterministic heuristic, the utility of a camera in FOV loss recovery depends not only on the covered FOV but also on the likelihood that it is available to participate to the selected covering solution. The likelihood depends on both the probability of the camera to transmit its data within the required time limit and the probability of the complementary cameras to also deliver their data too. For example, in Figure 6.13(c), camera S_1 provides a significant FOV coverage, and hence should be selected according to the priority function of the previous heuristics. Moreover, cameras S_1 and S_2 offer the best overall FOV coverage. However, if camera S_2 is not available then the pair S_3 and S_4 gives a better overall coverage than camera S_1 alone, even though S_1 has the largest FOV coverage among the three cameras.

For any camera Si, the probability $p_{j,unav}$ of a camera S_i ∊ neighbors(S_i) to be unavailable for the FOV loss recovery of camera *Si* is as follows:

$$P_{j,unav} = P_{j,obst} + P_{j.comm}(1 - P_{j,obst}) \tag{52}$$

Probability $p_{j,obst}$ is the probability of camera S_j being obstructed by a vehicle, and probability $p_{j,comm}$ is the probability of having a communication loss. Probability $p_{j,comm}$ is evaluated locally by S_i for every of its neighbors. Probability $p_{j,obst}$ is computed by every camera S_j , and communicated periodically to camera S_i.

The priority function of a camera S_j , called Expected PWC (EPWC), is defined as follows:

$$EPWC_j = PWC_j \times p_{j,part} \tag{53}$$

where $p_{j,part}$ is the probability of camera S_j to participate to a covering solution. It depends on the probability of the other cameras expected to participate in the recovery to also make their data available. This probability is estimated as follows:

$$p_{j,part} = \sum_{S_j \in Cover_m} \prod S_{k \in Unav(Cover_m)} (1 - p_{k,unav}) \tag{54}$$

where $Cover_m$ denotes the coverings that camera S_j is part of. Set *Unav*($Cover_m$) describes the cameras that are part of covering $Cover_m$ but did not make their data available yet. In the proposed heuristic, the sets $Cover_m$ are approximated by storing the *M* most frequently used coverings.

6.8. Experiments

The proposed methods were implemented as Java programs, and simulated on a PC desktop computer. Experiments studied the FOV loss recovery in the presence of moving vehicles, and for different traffic scenarios.

Figure 8.18. illustrates the case study used to model a real-life urban traffic situation. The white areas indicate the FOVs of the deployed cameras. Experiments considered three levels of redundant FOV coverage:

(i) minor redundant coverage (middle group),
(ii) moderate redundant coverage (left group), and
(iii) strong coverage (right group).

Moreover, the traffic scenarios included one, two, three, five, and fifteen cars moving at different speeds. The case study used the following parameters: the length of the monitored route was 70 meters with 4 lines (as seen in Figure 6.7.), each line being 3 meters wide. The vehicle speed was varied in the range 30 km/h to 60 km/h. The camera model had the following parameters (see Figure 6.6.): DMAX was 30 meters, and angle α was 40 degrees. The distance between the camera sensors and the monitored route was set between 2 and 6 meters. The simulated time was 8 seconds and the simulation step was 0.3 seconds.

Table 8.1. presents the total coverage loss without and with the proposed deterministic heuristic algorithm (first heuristic). Columns two and three are the number of cameras and cars in each traffic scenario, column three is the area of the coverage loss without any FOV recovering, and column four is the area of the FOV loss after using the proposed FOV recovery method. Column six reports the percentage FOV loss recovery of the heuristic as compared to no recovery being performed. The average FOV loss recovery is 52% for the nine cases, the lowest value being 19% and the highest being 80%. In all but one case, the FOV loss was about 40% or greater.

Tables 6.2. and 6.3. offer more insight on the algorithm's performance, including scalability for increased number of cameras and monitored vehicles. Table 8.2. refers to the case if no FOV loss recovering algorithm is used. Table 6.3. is for the deterministic heuristic. Columns one and two indicate the number of cameras and vehicles. Column three shows the average FOV coverage, Column four the minimum FOV coverage, and Column five the FOV coverage variance over time. Column six reports the execution time of the algorithm.

6.18. Case study for FOV loss recovery

# Cam. (1)	# Cars (2)	Total Cover. Loss (3)	Total Cover. Loss with Method (4)	Average FOV Loss Recovery (%) (5)
3	1	344.40	162.89	52
3	2	1386.53	598.42	56
3	3	1974.93	1002.39	49
4	1	717.22	201.25	72
4	2	1662.46	320.15	80
4	3	1447.28	468.89	67
5	1	382.42	306.89	19
5	2	736.67	445.05	39
5	3	1194.21	749.04	37

6.1. Total coverage without and with proposed method

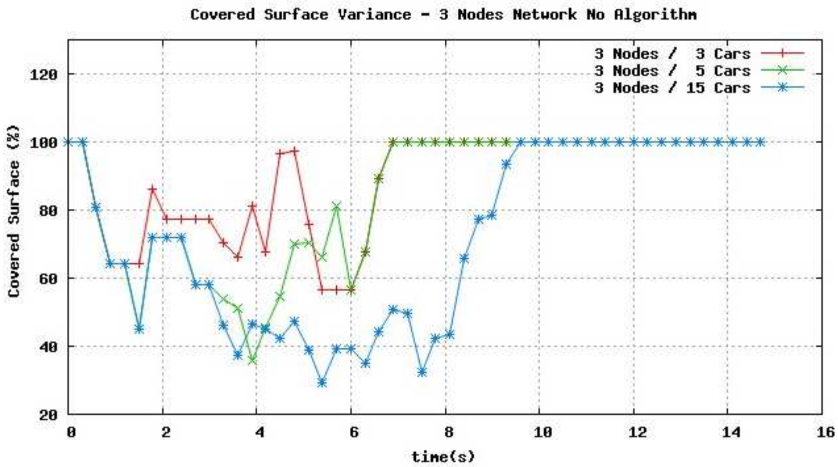
# Cam. (1)	# Cars (2)	Average FOV Coverage (%) (3)	Min. FOV Coverage (%) (4)	Variance (5)	Exec. Time (sec) (6)
3	3	84.45	56.67	2.52	0.99
3	5	80.53	35.89	4.69	1.14
3	15	63.8	29.31	9.34	1.23
5	3	89.01	18.11	4.57	1.18
5	5	84.58	18.11	6.63	1.31
5	15	72.21	7.36	14.40	1.56
15	3	90.28	57.16	2.18	3.14
15	5	86.13	9.7	4.31	3.31
15	15	65.13	15.85	10.43	4.01

6.2. Total coverage without proposed method

Table 6.1. (no FOV loss recovery algorithm used) shows that the FOV coverage loss increases significantly as the number of cars is higher. For 5 cameras and 15 cars (row six) the minimumFOV coverage can be as low as 7.36%. Column five indicates that the resulting variance is large, and hence FOV coverage is unreliable even in situations in which the FOV coverage is reasonably high, e.g., 5 cameras and 15 cars. According to 6.3., the proposed heuristic improves significantly the average FOV coverage, the minimum coverage, and variance for all cases. The average FOV coverage is above 85% in all cases as compared to 63% without algorithm. Also, the minimum FOV coverage does not drop below 65%, which is a significant improvement over the results in column four in 6.2. The variance is low (Column five in Table 6.3.). Hence, the algorithm reliably covers the FOV over the entire period. Column six shows that the algorithm scales well for larger examples. It is fast even for higher number of cameras and monitored vehicles, e.g., 15 vehicles and 15 cameras.

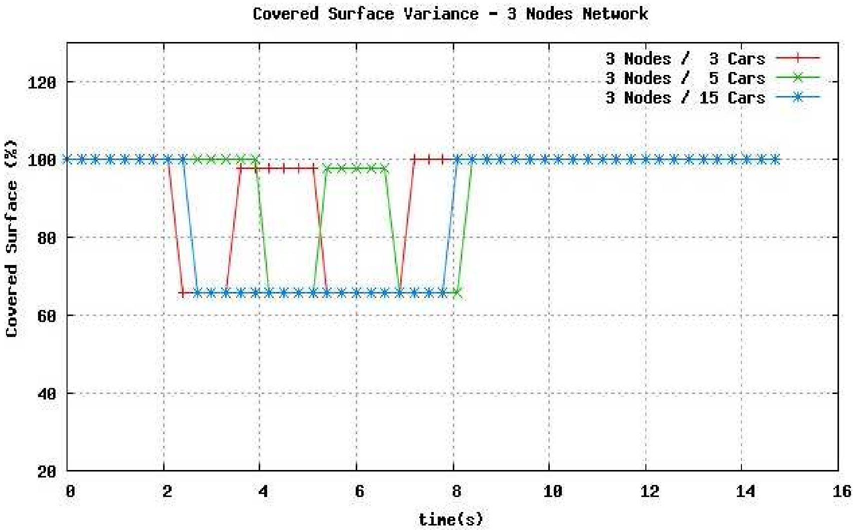
6.3. Total coverage with proposed method

Figures 6.19., 6.20., 6.21. and 6.22. give more insight about the covered FOV areas with and without the deterministic heuristic. Figure 6.19. and 6.20. correspond to a cluster with three cameras (the middle cluster in Figure 8.18.), and Figures 6.21. and 6.22. are for the more dense cluster with five cameras (the right cluster in Figure 6.18.). The plots present the variation of the percentage of the covered FOV area over time for the three traffic scenarios (three, five and fifteen moving cars). Coverage of 100% indicates that there is no FOV loss. Note that the deterministic heuristic is capable of providing FOV covering above 65% and close to 100%for most of the time. In contrast, the FOV loss can be close to 60%for long intervals of time (almost 50%of the time in Figure 6.20.), if no FOV loss recovery is utilized. This motivates that the addressed problem is important for comprehensive and continuous data collection in traffic management applications.



6.19. FOV coverage for 3 cameras without proposed method

The four plots suggest the following dependency of the FOV loss due to moving obstacles.



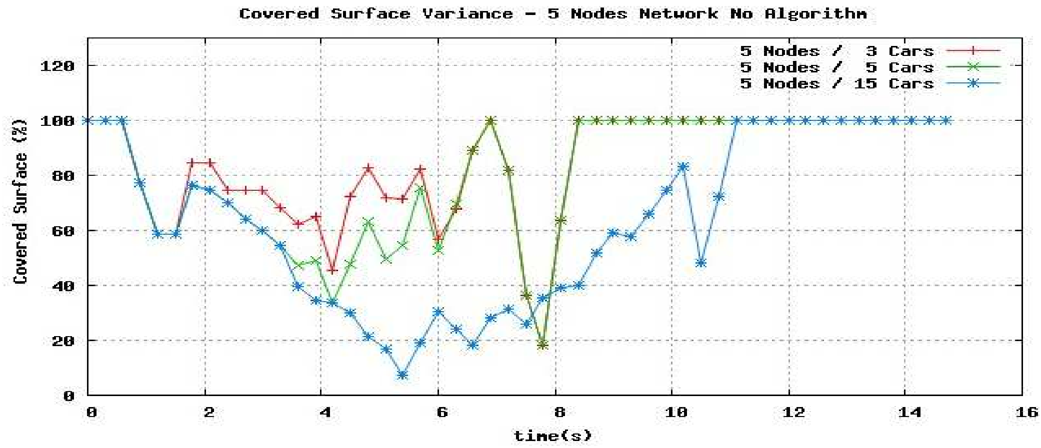
6.20. Covered FOV area for 3 cameras with the proposed method

Plateaus in the FOV coverage plots correspond to situations in which a moving vehicle obstructs uniformly a camera without another camera being able to cover any of the loss. Ideally, the size of plateaus should be very small. Identifying deep plateaus is important because they indicate poor deployment of cameras, and provide information to a camera deployment strategy on to where new cameras should be placed. Note that the plateaus in Figure 6.22. are narrower and shallower as compared to those in Figure 6.21., meaning that there are short periods of significant FOV loss. In Figure 6.21. (no recovery method), there is a deep plateau between moments 1 sec and 10 sec. Similar observations exist for the results in Figures 6.19. and 6.20.

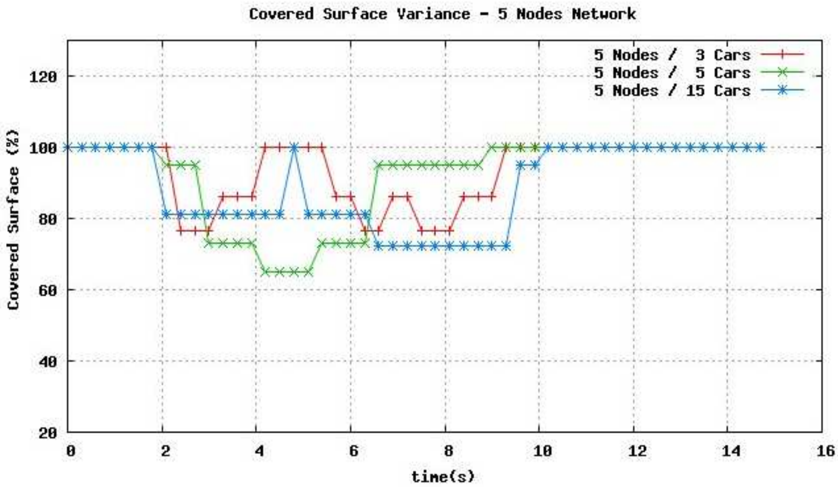
A spike in the FOV coverage plots indicates that either a car has quickly passed very closely to a camera, hence its FOV obstruction in time is short, or that a camera was able to cover the FOV loss, thus, bringing the coverage close to 100%.

If a spike is deep then the chances of losing more monitored information are greater. If the spike is shallow but broad, then the chances of not monitoring a car are smaller, but if this occurs then the time the car is not covered is actually longer. The spike width is related to the minimum speed of a car that might pass through the area without being identified by the monitoring system, thus offering information on the speed restrictions that could be introduced to improve traffic safety in populated area or other high risk zones.

Figure 6.23. shows the FOV coverage if static obstacles are present, such as stopped cars. The experiment considered two stopped cars for the case with 5 cameras and 5 vehicles. Without using the proposed algorithm, the FOV coverage is mostly between 60-70%, at times dropping to 40%. The proposed algorithm improves the FOV coverage to above 60%while keeping the FOV coverage to 100% for more than half of the time.

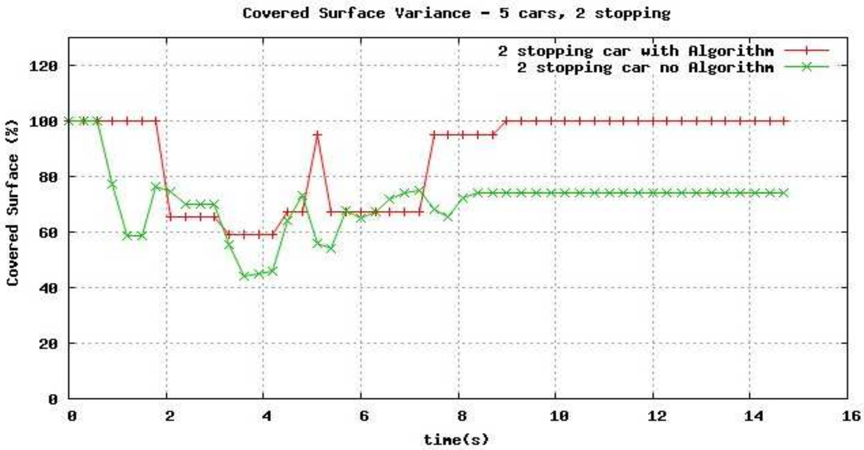


6.21. Covered FOV area for 5 cameras without proposed method



6.22. Covered FOV area for 5 cameras with proposed method

The next experiment evaluated the deterministic heuristic for a street intersection. Figure 6.11 illustrates the layout of the intersection. The simulation considered a segment of 100 meters for each of the two intersecting streets. The lane width was set to 3 meters. Four traffic lights were positioned as shown in the figure. The time set for the red and green light was 5 seconds. The experiment studied the impact of the number of camera sensors on the resulting FOV loss recovery.



6.23. Covered FOV in the presence of static obstacles

Two cases were considered: (i) monitoring the moving of 4 cars through the intersection by using 4 sensors, and (ii) monitoring 4 cars by 12 sensors. For the second case, 4 sensors were positioned at the same places as the sensors of the first example while the additional 8 sensors were distributed equally in the close neighborhood of the initial 4 sensors. Same traffic conditions were used in both cases. The obtained simulation results are plotted in Figure 6.20. Figure 6.24.(a) is for the 4 car - 4 sensors situation, and Figure 6.24.(b) for the 4 car - 12 sensors case. In both cases, the heuristic improves the final FOV coverage. Even though the FOV loss recovery improves with the increasing of the number of camera nodes, the amount of FOV loss recovery depends significantly on the positioning of the sensors. FOV loss recovery results only if there are redundant nodes available that can cover the areas lost due to the moving vehicles.

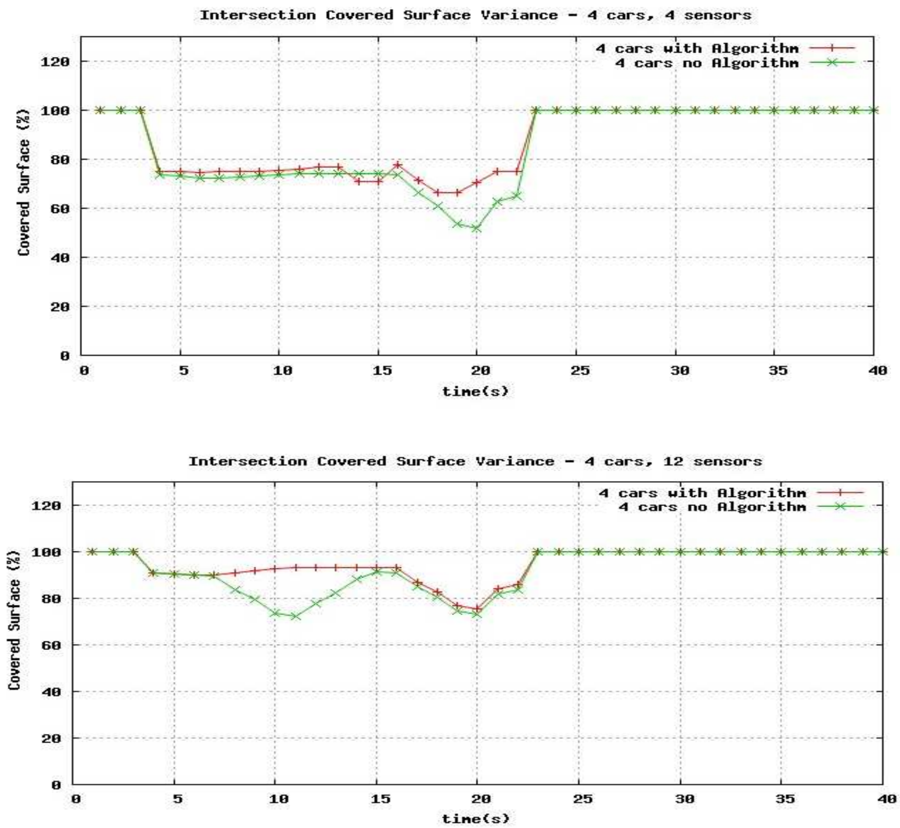
# Cam	# Cars	Proposed algorithm			LP-based algorithm		
		Average FOV coverage (%)	Min. FOV coverage (%)	Exec. time (sec)	Average FOV coverage (%)	Min. FOV coverage (%)	Exec. time (sec)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
3	3	98.17	75.98	1.18	98.30	80.09	12.21
3	5	93.365	83.74	1.21	97.26	80.01	12.23
3	15	90.52	75.98	1.35	85.84	55.86	12.30
5	3	98.88	74.62	1.27	98.57	80.09	12.22
5	5	98.40	74.62	1.53	97.06	80.01	12.24
5	15	90.52	60.25	1.68	82.41	60.25	12.27
15	3	99.26	90.54	3.42	99.08	84.46	13.66
15	5	96.83	89.28	3.48	98.27	84.11	13.77
15	15	92.49	72.54	4.27	87.17	74.57	13.79

6.4. Total coverage with proposed method

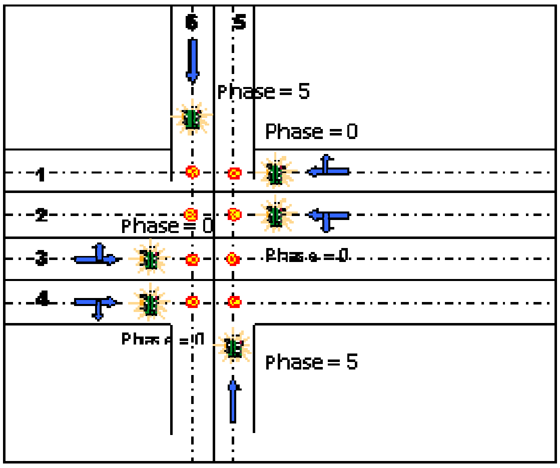
Table 6.4. compares the deterministic heuristic with the ILP-based method. For the two methods, the table indicates the average and minimum FOV coverage and the execution time for various number of cameras and vehicles. As expected, the ILP-based method offers better results in about half of the instances but at the

expense of a much longer execution time. Without powerful computing resources, the ILP-based method cannot be used for real-time traffic monitoring. In about half of the instances, the heuristics finds better coverage due to the way the selection procedure is implemented. The ILP algorithm has a tendency of turning off more cameras, especially if many cars produce large FOV losses. This is due to linear formulation, which enforces a camera to be turned off once its FOV is significantly reduced by an obstacle. The heuristic method might still turn on a camera with its FOV covered by other cameras, if these are obstructed by vehicles. This constraint is hard to express as an ILP constraint.

The last experiment evaluated the performance of the second heuristic, which is based on the stochastic formulation. The experiments consider that randomly only 80% of the cameras can communicate with each other while 20% are unavailable. The amount of acceptable FOV overlapping between two cameras was varied. The model in assumes that a camera is turned off if its FOV is covered k% by another camera. The values of parameter k were set to 30%, 50%, and 70%. This parameter controls the cameras that are candidates to be included into sets $S_{i,F}$. A larger value of k increases the cardinality of the sets. Two cases have been experimented for each case: (i) when only a restricted set of cameras can respond due to the set timing constraint, and (ii) when all cameras can respond. The second case is ideal and was used as a reference.



6.24. FOV loss recovery for the street intersection (a), (b) for example in Figure 6.25.

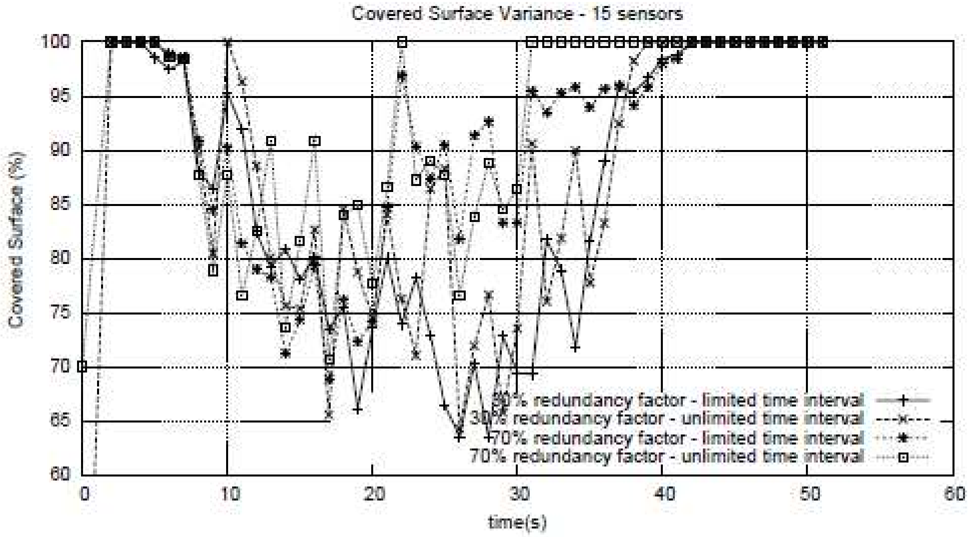


6.25. Layout of the street intersection example

Table 6.5. summarizes the resulting coverage over time and Figure 6.26. illustrates the coverage. The resulting coverage increases with the value of parameter k, even though there are small fluctuations due to the stochastic availability of the nodes. This experiment also suggests that selecting parameter k about 50% offers a good trade-off between the resulting coverage and the camera resources that are saved by being turned-off.

Time	30% redundancy			50% redundancy			70% redundancy		
	restricted	all	difference (%)	restricted	all	difference (%)	restricted	all	difference (%)
5	100	100	0	100	100	0	100	100	0
10	86.45	80.44	-7.0	80.44	84.46	4.75	84.46	78.91	-7.03
15	80.80	75.64	-6.82	78.35	74.34	-5.39	71.25	73.65	3.25
20	66.10	78.74	16.05	77.49	74.17	-4.47	72.26	84.92	14.90
25	72.94	86.42	15.59	82.19	87.33	5.88	87.33	89.03	1.90
30	72.83	65.75	-10.76	65.75	83.31	21.07	83.31	84.50	1.40
35	71.77	89.95	20.21	90.81	95.81	5.21	95.81	100	4.19
40	96.67	100	3.33	100	95.77	-4.41	95.77	100	4.23
45	100	100	0	100	100	0	100	100	0

6.5. Total coverage with stochastic heuristic

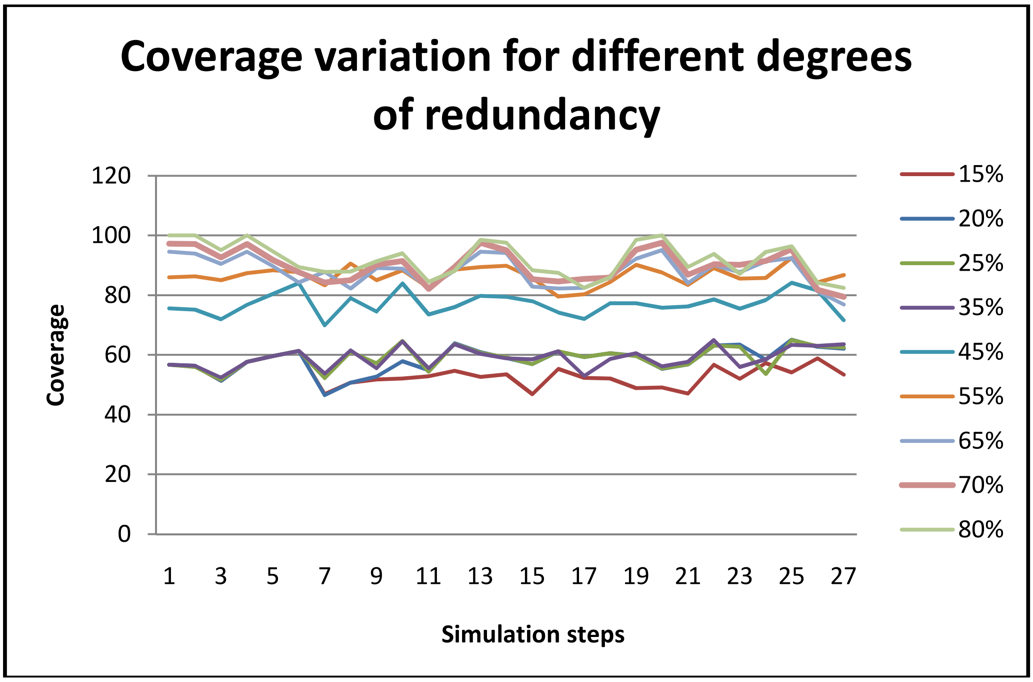


6.26. FOV coverage for the stochastic heuristic

The table lists the relative difference between the reference case when all available cameras can communicate, and the case when only a restricted set communicates due to the imposed timing constraints. The worst case difference is about 21% but for most of the cases it remains below 15%. Large differences occur if the additional cameras in sets $S_{i,F}$ can cover only small areas of the lost FOV. But for most of the cases, three additional cameras offer sufficient loss recovery. Finally, in some situation, the restricted case offered better recovery values than the unrestricted one. This is explained by the way in which cameras become randomly available during recovery.

6.9. In Depth Motivation

The proposed algorithm is used for traffic surveillance. We use dynamic FoV recovery in order to improve traffic surveillance algorithms. One of its key concepts is redundancy. In this case, we considered that two cameras are redundant if their FoV is overlapping more than a percentage that can be established. In our study cases, we considered this percentage to be 70%. Figure 6.1. presents the fact that at 65% the coverage is stabilized. The difference between higher or lower values would mean more loss in coverage or higher consumption of energy by more sensors turned on.



6.1. Coverage variation for different degrees of redundancy

The need for computing redundancy comes from the fact that for a good coverage, cameras have to be placed closed to one another. Even so, if the objects of the surveillance, in our case, vehicles, are becoming obstacles and obturate the field of view of the sensors, the coverage drops considerably. This is a major reason for using more cameras for traffic surveillance. When the FoV of one of the cameras is obstructed, other cameras that have the same FoV can monitor from another angle the lost FoV. For a good coverage, the redundancy needs to be high, but using so many cameras implies high costs and relatively short lifetime for the network.

The proposed algorithms address this issue from several perspectives such as mathematical point of view by heuristic model, proposing a traffic model for simulation and of course by simulation, including Monte Carlo simulation.

6.10. Conclusion

This chapter proposes two novel heuristic methods for real-time, distributed image acquisition through a network of traffic monitoring cameras. The goal is to minimize the FOV loss of the cameras due to dynamic obstacles by identifying the best set of additional cameras that can compensate for the loss. The set is identified under the timing and sampling constraints of the application and with the objective to reduce the utilized resources. The FOV loss changes dynamically depending on the traffic conditions. As wireless communication can be unreliable, the availability of a camera is also considered in the chapter. The two heuristic methods employ different cost functions for selecting the additional cameras used for FOV loss recovery. The cost functions are based on a new stochastic model for traffic monitoring, including the dynamics of mobile obstacles, unreliable communication, and resolution and timing constraints. The first cost function, Parameter Weighted Contribution (PWC), addresses deterministic situations by capturing the trade-off between the quality of recovery and the imposed timing constraints. PWC expresses the utility of a camera in FOV loss recovering, the available resources of a camera, and the capability of recovering multiple FOV losses of neighboring cameras. The second cost function, called Expected PWC (EPWC), addresses unreliable networks, such as wireless connections. EPWC extends PWC by incorporating the probability of a node to participate in FOV loss recovery, including the chances of the camera being obstructed by obstacles, experiencing data loss during communication, and other cameras used in the covering solution being also available.

The average FOV loss recovery of the deterministic heuristic is 52% with actual values between 19% and 80% of the lost FOV. The algorithm delivers FOV coverage of at least 63% for cases in which FOV coverage drops to only 7% if the algorithm is not used. The resulting coverage is close to 100% for most of the time while without the recovery to coverage drops to about 60% for about half the time. The heuristic is capable of improving the reliability of the recovered loss as motivated by the small variance of the solutions. The two heuristic methods are fast, thus scale well with the number of monitored cars and cameras. In contrast, the ILP-based method is much slower, and cannot express nonlinear or stochastic aspects. For time-constrained, unreliable communication, the stochastic heuristic offers a coverage that is only about 15% less than if communication is unrestricted. Also, turning off cameras that have about 50% of their FOV covered by other cameras offers good FOV loss recovery while saving resources.

7. Improving FoV Coverage Preservation Through Traffic Prediction

7.1. Abstract

This chapter presents an innovative idea of combining the micro traffic with the macro traffic. Micro traffic is represented by the fact that the behavior of each car influences the management of the algorithms in order to maintain a good coverage and the macro traffic is represented by the idea of events. If the road is fully loaded with vehicles, all the vehicles on a lane if they are close enough to one another are seen as a single event. An event is represented by one or more vehicles that are close enough to each other so the distance between them would be insignificant from the coverage point of view. The events form dynamically when a vehicle surpasses the minimum distance to another vehicle in order to form an event. Also, an event splits dynamically when a vehicle gets out of the range for which the distance is too big to be considered negligible. The modality in which the events are formed and split is largely presented above.

Another innovative idea is the fact that this algorithm uses the concept of prediction. It is important due to the fact that sensors stay off for a predicted period of time. The prediction is made taking into consideration the speed of the events and the data that has been sent by the sensors in the immediate proximity.

7.2. Introduction

Several attempts regarding traffic models have been done, but there are still plenty of improvements that can be added. In general, for modeling highway traffic, Gaussian densities [88] are used. It is difficult to model the connections between variables using Gaussian densities, thus, in the mathematical model with Gaussian densities, the variables of the system are considered independent and behave consequently. This approach alienates the results from the reality. This model works well if the highway traffic is light. In this case, the interaction between vehicles is small and the behavior of vehicles may be considered independent of traffic conditions.

Another approach, popular for traffic modeling, is the conditional autoregressive model. The idea behind this model is based on local probabilities only. In other words, this model is a car following one, where the dependencies between variables are influenced only by the adjacent traffic conditions. This model uses the Markov property [89]. In general, the concept of adjacently is considered to be represented by the situations from a certain segment defined as s. According to [90], the model assumes that the volume y observed at a location s obeys the formula:

$$y(s) = \epsilon_s + \sum_{r \in N(s)} \theta_r^s y(r) \tag{55}$$

where N(s) represents the neighborhood of s, ϵ_s is considered additive noise and θ is a parameter calculated with ridge regression procedure [91]. The authors of [90] try to overcome the Gaussian densities drawback using Bayesian networks [92] that allow a certain degree of dependences. Furthermore, the mathematical model proposed is assumed to learn from the given data sets that are divided into training and testing sets. Their results show the improvement compared to the Gaussian initial model, but still, the proposed method addresses only to car following models, ignoring the probability dependences with the whole traffic scenario.

In this chapter we present a mathematical model for traffic prediction tested by Monte Carlo simulation. Monte Carlo simulation is detailed further in the report. In this model we consider cars as events. An event can be formed by more than one car. If twoor more cars are on the same lane, have the same speed and the distances between them are less or equal to the minimum safe distance between two cars, those vehicles are considered part of the same event.

In our previous work [93, 94] we presented an algorithm for coverage preservation in the presence of dynamic obstacles, in our case, cars. Sensors decided for themselves if they are obstructed in such a proportion that they became unuseful and turned their camera off. Before turning off, they searched in their redundancy group the most redundant camera to turn on in order to maintain a certain degree of coverage. The sensor that was initially turned off due to the obstruction was immediately turned on after the car (dynamic obstacle) has passed.

This algorithm is an optimization of our previous work due to two important factors:

- Sensors consider events as being the dynamic obstacles
- Sensors stay off all the predicted obstruction period

These facts have as a result an optimization in power consumption. The sensors that turn off due to obstruction will turn on again only after the whole event has passed.

7.3. Overview

The dynamic of traffic is a complex issue and in time there were different ways to solve its formalization. The similarity between fluid dynamical approach and traffic flow is debated in [95]. The authors of the paper researched the behavior of normal traffic in comparison to laminar fluid behavior and turbulent traffic in comparison to fluid bottleneck situations. Their approach is to use Monte Carlo simulation in order to get as close to reality as possible, but for the mathematical simplicity of the computation, some variables such as the lengths of the vehicles are ignored and considered fixed. The model uses probabilities for determining the vehicles' speeds. This chapter offers a simulation model for localized traffic situations and a modality of coming back to a normal traffic situation after a bottleneck by observing the propagation of certain traffic state. The authors of [96] propose a car following model based also on the fluid dynamics accordingly defining a relation between speed and density. They establish the acceleration and deceleration with respect to the relation of the cars nearby. A new variable is also proposed and it refers to negative reaction that is a psychological factor, ignored in most of mathematical models. This variable finds its correspondence in the fluid mechanics as the viscosity. They start from the stochastic decomposition theory that determines a formula for the number of vehicles on a link (segment) X:

$$X = X_{\varphi} + Y \tag{56}$$

where X_{φ} represents the stationary number of vehicles on a link in uninterrupted traffic and Y represents the additional vehicles on a link as a result of traffic incidents. The variance of X is determined in [97]. They reach the conclusion that the average probability of having vehicles on a link is 25, without considering incidents. In the proposed model, the authors of [98] determine the congestion factor Ai:

$$Ai = \frac{V_n}{V_{free}} \tag{57}$$

where Vi is the vehicle speed based on the free speed, Vfree when there are totally n vehicles on a link. Reference [98] also gives a thorough description of the evolution of the mathematical models for traffic from its early beginnings. We consider relevant to mention a few theories regarding this subject. The authors of [98] classify the classical traffic models into three major approaches according to the analogy made: microscopic traffic models based on particle behavior analysis, mesoscopic models based on gas kinetic behavior and macroscopic traffic models based on fluid dynamic behavior. The concept of microscopic approach is based on the acceleration and deceleration of the current vehicle with respect to the behavior of the nearby vehicles. This model is called the car following model and was first introduced in [99]. The model was improved by adding the safe distance concept that directly influences the velocity of the vehicles was introduced in [100]. The mesoscopic models compare the interactions between particles of a gas to the interactions of vehicles on a road. Based on this idea the authors of [101] mathematically model the concept of acceleration and overtaking behaviors and obtain the critical density of the phase transition from free flow congestion. Since then, the model was improved by also modeling the conditions in bottleneck situations. References [102, 103] propose models for estimating the travel traffic delays also considering the congestion probability. Based on these studies, the effects of introducing traffic lights in different intersections were analyzed from the decongestion and waiting time perspective. The comparison of traffic behavior with the behavior of fluid dynamics is included in the macroscopic traffic theory. The idea of this approach is based on the average factors such as velocity, density, etc. Starting from the fundamental diagram describing an uninterruptible traffic system [104] the model was developed by adding the correlation between velocity and density and the viscosity [105] property that describes the reaction of the driver to the events in traffic.

The authors of [106] debate the capacity of a freeway. They also try to determine the maximum number of cars with respect to their density and speed.

The problem they try to solve is the maximum throughput $Q(\rho)$ of the freeway for which the car speeds are not influenced, where the mathematical definition for the freeway throughput

$$Q(\rho) = \rho V(\rho) \tag{58}$$

where ρ defines the density of the cars on the freeway and $V(\rho)$defines their corresponding velocities. The approach proposed by [106] treats the case of a single lane and the model is a car following one. In this model, the authors treat time in a discrete way and they debate on the discretization step that inevitably implies a certain loss of information. The model proposed in [106] has the basis in the Krauss model [107] that is proved to be free of collisions. Real traffic is not collision free, so the approach presented in [107] shows an unrealistic mathematical model for traffic. Basedon that model, the authors of [106] consider the acceleration used in their model 5,4km/h per second, the deceleration 16,2 km/h per second and the maximum velocity is considered to be 81 km/h. Even if the model is a car following one which means that the behavior of the current car is adjusted with respect to its adjacent cars' behavior, [106] presents three traffic behavior types: laminar traffic in which each car drives with its desired speed due to the big distance between the cars. An intermediate traffic behavior is the coexistence one in which a distinct group of cars are in a jam traffic and the others are in a laminar situation. The last case is the jam traffic situation that is defined by [106] as being a sequence of adjacent cars driving with speed less or equal to $v_{max}/2$. The cars between two neighboring jams are considered to be in laminar flow. The mathematical model proposed in [106] was realized for one lane. As mentioned earlier, on the lane, the authors consider the possibility of different types of traffic such as laminar and jammed. In order to establish the portions of the lane that have a certain type of traffic, the lane is considered to be formed by cells. So, in their model the 1 way lane road is split into cells of length l_c, where l_c represents the length a vehicle occupies in the average in a jam, for example, $l_c = \frac{1}{\rho_{jam}} \approx 7.5\mathrm{m}$. The transitions between one traffic state to another is also debated from its similarity to the transition of a fluid in a gas state and from a gas state to fluid state. Initially a coagulation point is formed that increases more and more. In the state changing zone exists an equilibrium domain that has both liquid and gas characteristics. The authors of [106] say that this domain might assure the maximum traffic throughput. Some important conclusions reached in [106] are the fact that only stochastic models allow to look at meta-stable states, spontaneous transitions all which are important to real time traffic. Also, according to [106], traffic is best described by 1 phase model and this model has no theoretical justification. The breakdown prediction becomes feasible only for 2 phase or 3 phase models.

7.4. Motivation

Coverage preservation realized by a good sensor management is tight to their response to the traffic flow that represents the dynamic obstacles. As shown above there are several approaches to solve this problem. It still remains an issue especially if the purpose of the model is maintaining a certain degree of coverage.

The algorithm presented in our previous work [93, 94] is optimized. Its improvements are presented in the current chapter. A sensor stays off until the whole event passes, even though an event can be formed by several vehicles. Considering the conditions that have to be fulfilled in order for a car to be part of an event, it is obvious that turning the sensor on and immediately turning it off again would be a waste of energy with very little gain. The gain would be the visibility of the area between cars, but that area might already be covered by another sensor. Furthermore, the speed of the event can be high, so the area that might have been seen if the sensor would have been turned on would have been a short glimpse. Gathering the data in such a short time and process it, it is also difficult to accomplish. For these reasons we considered that this approach is an improvement.

Another significant advantage is the prediction aspect. Sensors still turn off when they are obstructed and they become unuseful, but they are now able to compute when the event would have passed and turn themselves on again. Proceeding this way, more energy is saved without significant loss in coverage.

7.5. Problem Description

Traffic prediction is difficult. This is no news and depending on the application, there are different modalities to predict traffic. There are two possibilities of analyzing this issue: from a micro or a macro perspective. The micro perspective means that the behavior of each car is observed. The result of this observation has to be understood as a hole in such a way that the behavior of traffic as a hole can be approximated. This is difficult and not likely to achieve. The other approach is to have a global view from the beginning. This allows understanding traffic as a flow. The majority of models see traffic as a fluid behavior.

The model we propose in order to have a good prediction is a mix between the micro and macro perspective. The idea behind this is that when sensors are turned on, they see cars and they register their speed, lane and orientation. We will explain how this is realized later in the chapter. The characteristics of cars mentioned above are registered in a global database together with their offsets. Taking into account the conditions, if they are fulfilled, events are formed. If a sensor is turned off due to obstruction, it looks in the database to see the characteristics of the event and it turns itself off on the period it predicts the event will by still obstructing for itself. After the predicted period passes, it will automatically turn itself on, again.

This method has both a distributed and a centralized component. The distributed component is the part in which the sensors decide if they are obstructed and become unuseful and turn themselves off the sensors compute the duration of the obstruction the sensors turn themselves on again after the event has passed

The centralized component is necessary because all the information that sensors compute, are registered on a server and all the sensors have access to that global database.

7.6. Sensor Capabilities

The method proposed has only been tested by different types of simulation, but the target is to implement and test this algorithm in practice. In order to do this the sensors must have some capabilities:

- sensors must have a video camera that has an optical focus capacity
- sensors must have an internal clock
- sensors must have wireless transmission capacity

Initially, after deployment, on the lanes of interest, that means on the lanes on which the sensors will monitor and register events, a special car will be driven that will help at sensors' calibration. This car will have a known constant speed that will emit the code corresponding to the lane it drives on. This car will also have a pattern drawn on it (for example two big spots - one color in front and another color at the back) in order to distinguish the direction of the lane. So, if a sensor recognizes the calibration car driving in its field of view, that sensor will have information about the lane and its direction. Furthermore, computing the data offered by the calibration car such as the lane, its direction, and also considering the velocity of the car, relationships regarding the relative positions between sensors can be determined.

The simplest methods that sensors could use in order to detect the calibration car and to recognize its pattern are phase detection and contrast measurement. Phase detection is similar with the way eyes form their image. Two points are needed and the image is than formed. Contrast measurement is based o computing the blur of the image. When the contrast between pixels is maxim, the blur is minim, so the image has the best quality. This method is easy to realize and the sensor can compute its blur when the calibration car drives in front of it. After adjusting its best image for each of the two lanes, it can automatically switch at those focus positions according to the lanes that have the vehicles on them.

7.7. Algorithm Description

When a sensor is on, it registers in a database all the cars that appear in its visual area. More specific, each sensor sends to the database the lane the vehicle is on, the speed of the event and the starting and ending points of the segments defining the car. The server then computes the properties of each vehicle and if the conditions are fulfilled, from separate segments defining particular cars, events are formed. If a car is driving at a certain distance from the other cars, that car will form an event by itself and the car the sensor had registered will remain a singular event until the conditions will be accomplished and will be coupled with another event.

If two events are on the same lane and the distance between the events is less then d_{min} (safe distance between cars), the events will be concatenated and the result will be one concatenated event. The extremities of the composed event are given by the minim and the maxim values on the OX axes from all the unique segments that are contained in the compose event.

We note f (front) the margin of a segment that has the greatest offset on the lane and we note b (back) the margin of the same segment that has the minim offset on lane. We consider that the offset is computed on the direction of event propagation. In this case we have:

$$\boldsymbol{f_E} = \mathbf{max}(\boldsymbol{f_i}),\ \boldsymbol{f_i} \in \boldsymbol{F_E} \tag{59}$$

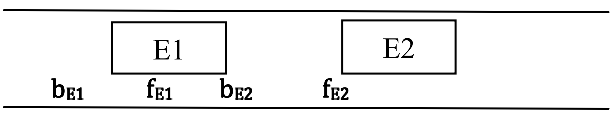
$$\boldsymbol{b_E} = \mathbf{min}(\boldsymbol{b_i}),\ \boldsymbol{b_i} \in \boldsymbol{B_E} \tag{60}$$

where

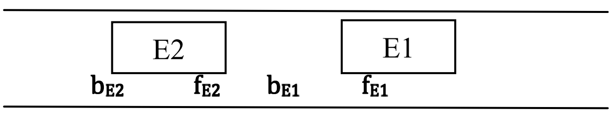
- f_E is the front of the event,
- b_E is the back of the event,
- F_E is the set of all front segments frontiers,
- B_E is the set of all back segments frontiers

Considering these notations, two events E1 and E2 will be concatenated in one of the following situations:

- $\boldsymbol{f_{E_1}} < \boldsymbol{b_{E_2}},\ \boldsymbol{d}(\boldsymbol{f_{E_1}}, \boldsymbol{b_{E_2}}) \le \boldsymbol{d}_{\mathbf{min}},$ (61)
  $\boldsymbol{v_{E_1}} < \boldsymbol{v_{E_2}}$ (62)
  ⇨ $\boldsymbol{E}\{\boldsymbol{b_{E_1}}, \boldsymbol{f_{E_2}}, \boldsymbol{v_{E_2}}\}$ (63)



- $\boldsymbol{f_{E_2}} < \boldsymbol{b_{E_1}},\ \boldsymbol{d}(\boldsymbol{f_{E_2}}, \boldsymbol{b_{E_1}}) \le \boldsymbol{d}_{\mathbf{min}},$ (64)
  $\boldsymbol{v_{E_2}} \ge \boldsymbol{v_{E_1}}$ (65)
  ⇨ $\boldsymbol{E}\{\boldsymbol{b_{E_2}}, \boldsymbol{f_{E_1}}, \boldsymbol{v_{E_1}}\}$ (66)



- There is one more concatenation situation when a car is on a separate lane and it changes he lane, coming into a colon of cars that form an event. In this situation the vehicle is just inserted into the event. Nothing else changes due to the fact that the extremities of the event remain unchanged. The speed v also remains the same because if the vehicle that was inserted into the event had to have the same speed as the event in order to be integrated in it.

Where:

- E is the new event resulted by the concatenation of the two,
- d(a,b) is the distance from point a to point b
- d_{min} is the minimum distance between cars

The events are concatenated if they are on the same lane and are kept in a common database that is available to all sensors (s). The events registered in the common database are constantly normalized to the current time tcrt by the propagation of the event in time. The propagation is computed on a certain segment with respect to the speed v of the event and to the difference in time from the last propagation of the same event. We note:

- t_{DB} the current time of the event in the database
- t_C the current time
- v_{DB} the speed of the event in the database
- o_{DB} the offset of a segment that is part of the event from a lase. This offset is useful at t_{DB}

By normalization we will update the offsets of all segments and in the same time the time of the last propagation of the event in the database in order to simulate the movement of the event in the database according to its speed v. The result will be:

$$\boldsymbol{t'_{DB}} = \boldsymbol{t_C} \tag{67}$$

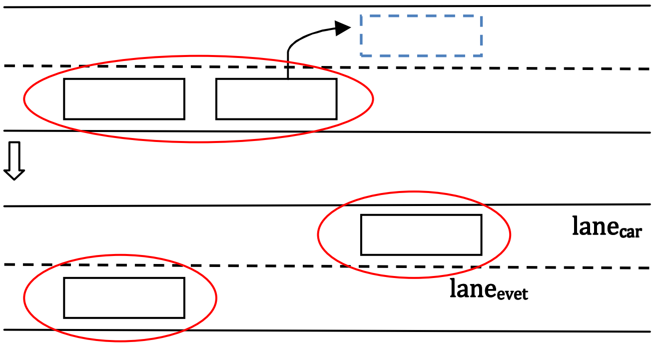
$$\boldsymbol{o'_{DB_i}} = \boldsymbol{o_{DB_i}} + (\boldsymbol{t_C} - \boldsymbol{t_{DB}}) * \boldsymbol{v_{DB}}, \forall\, \boldsymbol{o_{DB_i}} \in\ \boldsymbol{o_E}, \tag{68}$$

- o_E is the set of all segments' offsets from the event
- t'_{DB} is the new current time from the database
- o'_{DB_i} is the new offset of the segment from the event

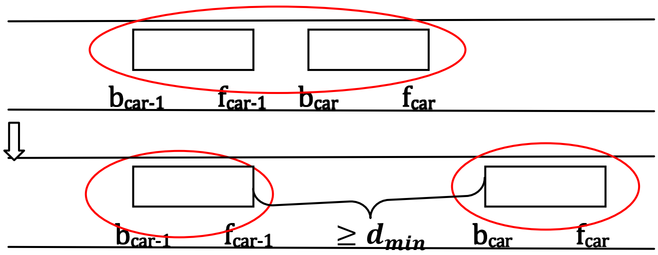
Once an event is registered in the database, for a simulation step, its lane or speed will not be modified. If a vehicle that was previously registered in an event is noticed outside an event, the situation is updated in the database with respect to the new real situation. From this update a modification of the segments' position in the event can appear or it is also possible that the segment is completely removed from the event, if the conditions are not fulfilled anymore. In this case it is possible that old event to be separated in other two distinct events if the distance condition is not respected anymore.

A separation of a vehicle from an event can take place in one of the following situations:

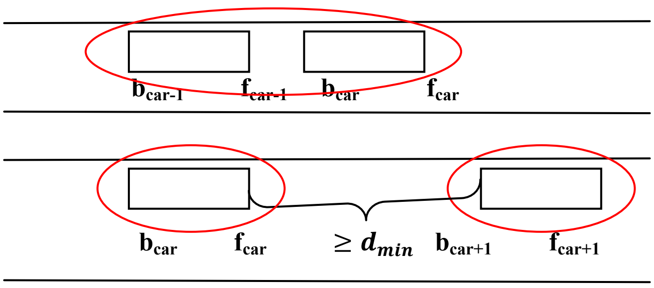
1. $\boldsymbol{lane_{car} \neq lane_E}$



2. $\boldsymbol{f_{car} = f_E}$ (69)
$\boldsymbol{v_{car} > v_E}$ (70)
$\boldsymbol{d(b_{car}, f_{car-1}) \geq d_{min}}$ (71)



3. $\boldsymbol{b_{car} = b_E}$ (72)
$\boldsymbol{v_{car} < v_E}$ (73)
$\boldsymbol{d(f_{car}, b_{car+1}) \geq d_{min}}$ (74)
Where



b_{car+1} and b_{car-1} have been noted the cars that are in front and after the considered car

We can distinguish the next possible situations:

Corresponding to 1. if:

- $\mathrm{d}(f_{car-1,}b_{car+1}) > d_{min}$ (75)

⇨ $\mathrm{E}_1\{b_{\mathrm{E}}, f_{car-1,}v_E\}$ (76)

⇨ $\mathrm{E}_2\{b_{car+1}, f_{E,}v_E\}$ (77)

b_{car+1} and b_{car-1} have been noted the cars that are in front and after the considered car

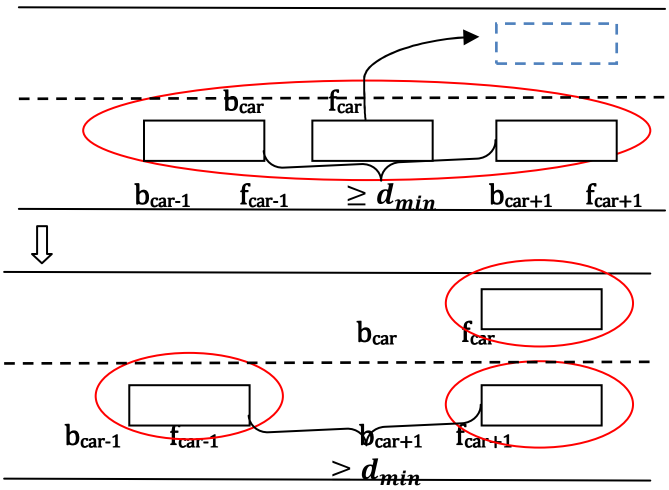
We can distinguish the next possible situations:

Corresponding to 1. if:

- $\mathrm{d}(f_{car-1,}b_{car+1}) > d_{min}$ (78)

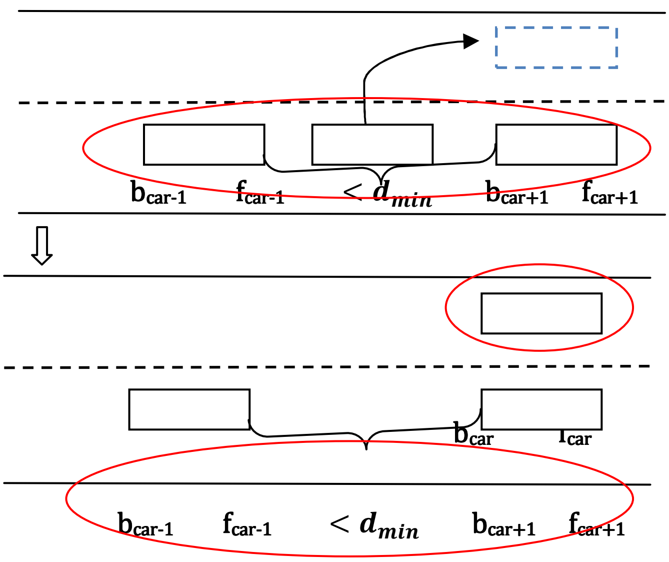
⇨ $\mathrm{E}_1\{b_{\mathrm{E}}, f_{car-1,}v_E\}$ (79)

⇨ $\mathrm{E}_2\{b_{car+1}, f_{E,}v_E\}$ (80)



- $\mathrm{d}(f_{car-1,}\ b_{car+1}) < d_{min}$ (81)

the event remains the same with the exception that the segment corresponding to the considered car is removed from the event



b) Corresponing to 2.

$$\Rightarrow \ E\{bE, f_{car-1}, v_E\} \tag{82}$$

c) Corresponing to 3.

$$\Rightarrow \ E\{b_{car+1}, f_E, v_E\} \tag{83}$$

To this event, a new event will be added $E'\{b_{car}, f_{car}, v_{car}\}$ corresponding to the considered car.

If a turned on sensor realizes that it is obstructed in a proportion greater than the maximum given obstruction, it is considered that the sensor becomes useless and it will turn off.

For all the turned on sensors, a coverage analysis is performed and if two sensors that are turned on and those sensors are redundant, the sensor that covers the less, will be turned off. The sensors that are off will be turned on again if two conditions are fulfilled:

- There is no sensor that has a better coverage that itself that is on and is redundant with it
- It is not obstructed in a greater proportion greater than the limit obstruction level by any events registered in the common database

Due to the fact that the sensors that are turned off, they cannot compute their real obstruction with respect to the vehicles that are in front of it. In this case, the obstruction will be computed taking into consideration the speed, the position and the length of the event.

A composed event is considered to be obstructing on all its length on the road because the spaces between the segments are too small at those speeds and the processing capacity of the sensors is too small in order to be able to register possible events that are not on the closest lane to the sensor.

An important concept that is used in managing sensors in the current algorithm is redundancy. As we mentioned earlier, two sensors that are redundant cannot be on in the same time. A sensor is considered to be redundant with another sensor if they cover the same are in a proportion greater than an established limit. This concept is more detailed in our previous work [93].

The mathematical model used in simulating the traffic decisions of cars like overtaking, speeding or decelerating, etc are the ones we developed and presented in [24]. In the current chapter we defined the model that describes traffic also from a global view. This way a more accurate simulation based on events and on the idea of event propagation is realized.

7.8. Performance

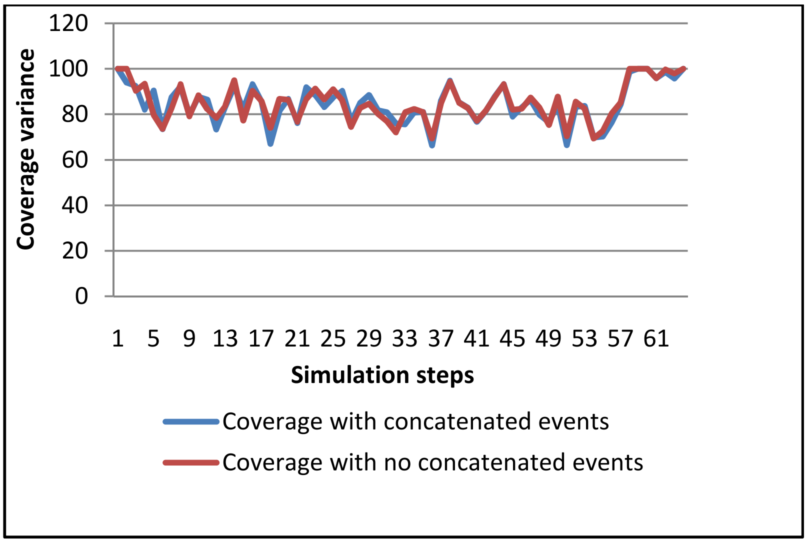
The experiment section shows the efficiency of this algorithm in comparison with the algorithm presented in Chapter 3. The current method was implemented as a Java program and run on a desktop PC. The algorithms compared in this chapter with the presented algorithm are the algorithm from [94], the situation where are no algorithms applied and the algorithm where the traffic is analyzed at micro level. This means that no composed events are considered.

The traffic cases were generated with a generator also implemented as a Java program. Traffic was generated according to real situations. The method used in simulating traffic is called Monte Carlo simulation. A more detailed description of the Monte Carlo simulation that was used can be found in the next chapter. Still it is important to emphasize the fact that each Monte Carlo simulation lasted for 1 day in order to analyze all traffic situations. The graphics presented below are snapshots of the worst situations found. In addition to this, we simulated how the current algorithm would work if we considered that sensors have limited battery. We did this by allowing the sensors a limited time in which each sensor could be turned on and perform traffic surveillance.

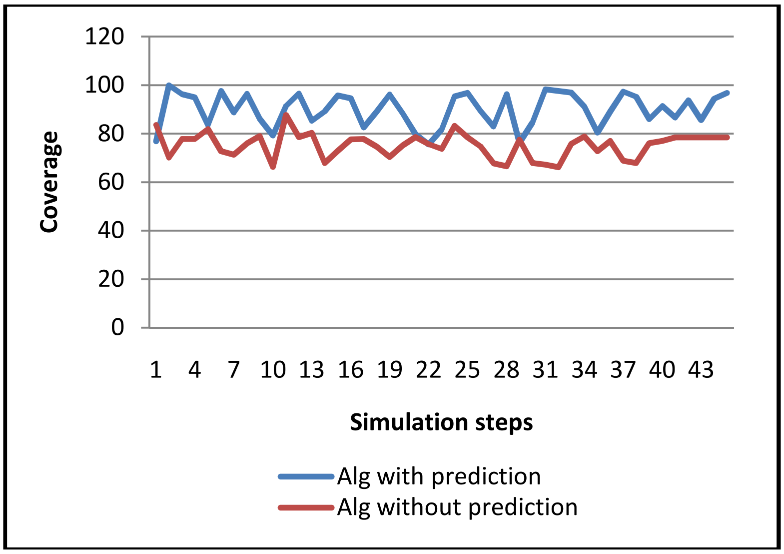
It can be observed in Figure 7.1. that the difference between the case when the cars are concatenated forming composed events and when the cars are not concatenated and each car is a single event is really small. As expected, when events are not composed, the coverage is better due to the fact that sensors turn on and off after each single event, meaning after each vehicle. It is important to notice that the minimum coverage value in the concatenated events case is 66.303% and in the case of single events is 69.363%.

Figure 7.2. shows the comparison between the performance of the current algorithm and the algorithm presented in our previous work [93, 94]. It was described the fact that not all sensors are turned on and the reason was shown. In the previous algorithm, the maximum number of sensors that were on at a time with no redundancy between them was computed at the level of redundancy groups.

The idea was that the sensors communicated between them in their redundancy groups in order to determine which sensor to turn on. Due to the fact that in a redundancy group, the distances between the sensors are relatively small, the energy spent on communication was not considered to be an issue. Furthermore, due to the fact that not all the sensors are on, the best coverage situation given by the maximum number of sensors on from each redundancy group is considered to be the optimal situation and was calibrated in our previous tests at 100%. The difference shown in Figure 7.2 comes from the fact that the calibration was computed with respect to the highest coverage value from both sets of data: the old algorithm and the new one.



7.1. Comparison between coverage levels with concatenated and no concatenated events



7.2. Comparison between the proposed algorithm and the old algorithm

It is obvious that the present algorithm performs much better than the older one. This comes from the fact that in the current algorithm, the maximum number of sensors that have no redundancy between them at the level of the network is turned on. This means that in the current algorithm, the number of sensors that are on at a time is higher and the coverage obtained is better.

Both algorithms have to constantly communicate with the central unit in order to provide real time data and also to request information about the status of the whole network, more in the current algorithm regarding the event prediction and in proportion also regarding the sensor management.

The comparison between algorithms was realized by normalization to the same value. Both algorithms computed their coverage performance and the results were normalized with respect to the unique value determined by the total coverage given by the network with all sensors turned on.

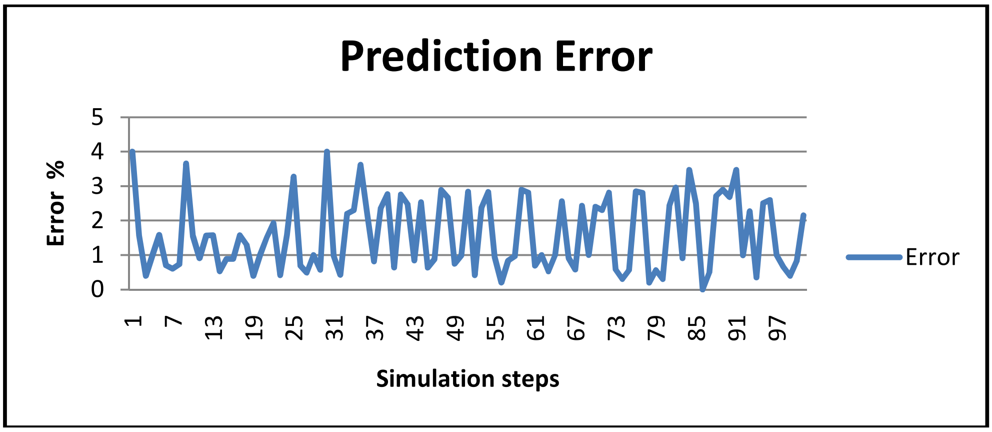
7.9. Prediction Error

The algorithm presented above is based on prediction. The prediction is used to compute the moment an event has passed the obstructed sensor and can turn itself on again.

The prediction error refers to the situation in which a car is not situated in reality where it would have been according to the prediction. The prediction error is determined by the fact that the obstructed sensor turned itself off when it became unuseful due to the events in front of it and is supposed to turn on when according to its prediction, the event has already passed. The error appears if the obstructed sensor turns on and the event has not passed its FoV.

The prediction errors can have several causes:

- when an event has entered the FoV of a sensor, but the event was not entirely seen by the sensor and the sensor is obstructed in such a proportion that has to turn itself off, the prediction for the event that has just entered its FoV is slightly to be accurate because the length of the event was not known by the sensor
- when a sensor turned itself off due to obturation by an or several events, it predicted their evolution on the lanes, but on the duration of the idle period for the obstructed sensor, the events might have split or changed lanes, or changed their speed, so the prediction can be altered.



7.3. Prediction error

In Figure 7.3. the prediction error for Monte Carlo simulation of the traffic from 9 am, situation presented from the coverage perspective in is shown Chapter 5.

7.10. Conclusion

The presented algorithms were developed as solutions to traffic surveillance using wireless sensors. Both solutions work and are perform well. Between the two algorithms there is a tradeoff between coverage and energy consumption. The level of energy consumption was not simulated but the remark comes from the fact that more requests to the central unit mean more energy consumption. We intend to test the two algorithms with respect to their energy consumption to see how big the tradeoff between coverage and energy is, but the object of the current chapter was to show the coverage performance of the new algorithm and we can conclude that the performance is high.

8. Monte-Carlo Simulation Of A Dynamic Coverage

8.1. Abstract

This chapter presents the proposed algorithms with their performances. The simulations that are performed are realized in accordance with the reality. The algorithms that are tested in order to validate their performance are presented as pseudo code. The detailed description is done in the previous chapters.

As mentioned above, for simulation real case scenarios were used. This type of simulation is called Monte Carlo simulation. The simulation performed has a mathematical foundation presented in Chapter 8.4. The mathematical model of the traffic simulator is described. The model takes into consideration both the saturated situation, when the traffic is jammed and also the unsaturated situation, where vehicles might travel with their proffered speed. The mathematical model also computes the time reaction of the driver, the minimum distance between vehicles that is considered to be safe, etc.

The simulations are performed in order to present the performance of the two algorithms: with and without prediction. They were also compared with LPSolve, which represents the mathematical version of the algorithm, using linear programming. The datasets used to perform Monte Carlo simulation were collected during some studies.

For relevance, tests are performed for different hours of the day to see how coverage varies with respect to traffic.

8.2. Introduction

This chapter investigates the benefits of Monte Carlo (MC) simulation in case of real-time distributed image acquisition through a wireless video-sensor network. The main goal is to prove the dependability of a coverage preservation algorithm designed for such kind of networks. The video-sensors provide image acquisition in the presence of dynamic disturbances, which obstruct the Field-of-View (FoV) of the cameras. The dependability of coverage preservation is not a trivial task to solve due to the diversity of dynamic interference in the FoV of the sensors that has a huge impact. Thus finding the worst case is a real challenge. One proposed algorithm tries to recover the area coverage by computing an optimized set of cameras based on redundancy. Another proposed algorithm is using prediction along with the redundancy concept. First, the problem is analyzed from a mathematical perspective. Then, the dependability of the algorithms is proved by experimental results through MC simulation. Test cases results consider real measurements from a road traffic surveillance system and are presented at the end of the chapter.

8.3. Argument

Real-time data acquisition from broad geographical regions is critical for many applications in transportation, infrastructure management, defense, homeland security, environmental and habitat monitoring, and agriculture [52, 58, 59, 60, 61, 62]. In spite of specific nuances, these systems are similar in that they must collect huge amounts of metadata, e.g., images, sound, temperature, toxin levels, etc., perform local processing, communicate and coordinate with each other through wired and/or wireless networks, and collaborate in achieving global and local goals. Their complex functionality is also subject to stringent performance and design constraints, like hard and soft timing deadlines, sampling and precision requirements, communication bandwidth, low power and energy consumption.

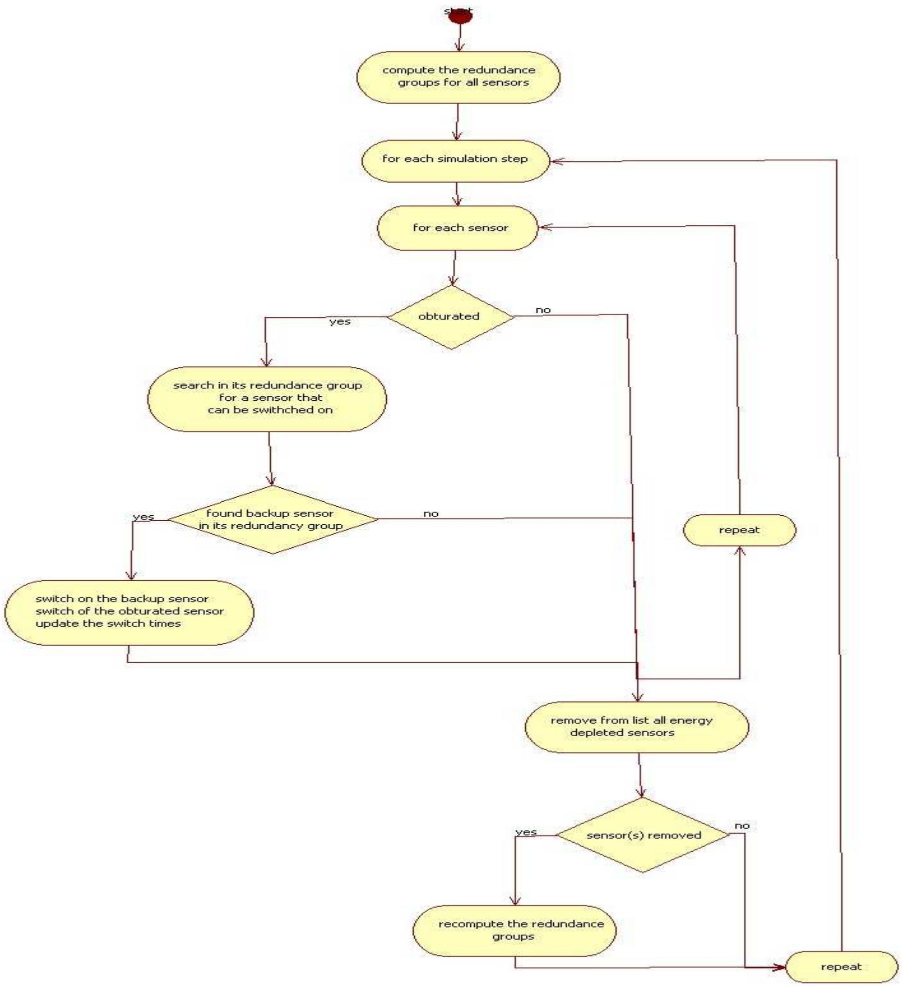
The problem addressed in this chapter regards the dependability of distributed algorithms for optimum coverage preservation in the presence of dynamic disturbances, different constraints (e.g. real time requirements – decision making) and nondeterministic factors (e.g. quality of communication in the case of WSN). Dependability reveals the degree of confidence the system can offer. Due to the fact that these algorithms are of NP complete complexity, they require heuristics in order to evaluate their performance. Often, in the evaluation of the algorithms, Monte Carlo methods are used in order to get as close to real results as possible.

An application that illustrates the problem discussed above is real time field of view (FoV) recovery in the presence of dynamic obstacles. To prove the dependability of the algorithm we then use a MC simulation. However, the dependability of a system is a difficult problem. What we address gets even more complex due to the variety of test cases that makes finding the worst case a complicated issue. What we try to determine by Monte Carlo simulation is the quality of FoV preservation in the presence of dynamic obstacles. Coverage preservation is achieved by real-time FoV recovery. A main issue here is the problem of dependability considering dynamic obstacles that obstruct the FoV. Indeed, they have as a consequence a decrease of coverage. In our previous work [94] we proposed a greedy algorithm that recovers in real-time the lost FoV. In this chapter we present an improved version of the algorithm and we determine the dependability of coverage preservation using MC simulation. We also present the pseudo-code for the algorithm presented in Chapter 8. At the basis of the MC simulation is a mathematical model for traffic analysis that helps in validation process. In order to obtain accurate results regarding the proposed algorithm, the MC simulation of the algorithm is also compared with an Integer Linear Programming (ILP) solution. As mentioned above, ILP techniques are usually used to resolve knapsack problems. In our scenario, we used ILP to determine mathematically the sensor management in order to obtain the best coverage by maximizing the coverage equation. However, due to scalability problem, the ILP solution is not suitable for practical applications. We detailed ILP in Chapter 8.

This work presents a method having a good performance in coverage recovery. It is then validated using Monte Carlo simulation, a popular method due their close probability of happening in real physical tests. This is why MC is used in various applications that include terrestrial or air traffic simulation [63]. Moreover, diverse reaction to unpredictable situations like volcano reactions and best plans to minimize the damages [71], forest crossing roads [68] and traffic [61] simulation are performed using MC.

8.4. Coverage Recovery Algorithms

8.4.1. Algorithm Without Prediction



8.1. Activity diagram for the algorithm without prediction

Figure 8.1. represents the activity diagram of the algorithm without prediction. The corresponding pseudo code is presented in Figure 8.3.

```
for (*each camera from set s) {
    *determine the FoV intersection with the road;
    *compute common road coverage;
    if (common coverage > k)
        *add that sensor to s' redundancy group of
    sensors;
    *sort the s' redundancy group of sensors descending;
}
```

8.2. Pseudo-code for computing the redundancy groups

Each of them then establishes if the degree value of the intersection is greater than *k*. We consider *k* as the percentage of common useful intersection between the FoV of two cameras in order for the cameras to be considered redundant. All sensors that satisfy this condition are then added to the initial sensor redundancy group. Finally, each redundancy group orders the sensors descending with respect to their useful FoV. The pseudo-code for computing the redundancy groups is presented in Figure 8.2. Choosing the camera that can best recover the lost FoV is resented as a pseudo-code in Figure 8.3.

```
if (a car enters in the FoV of a camera s) {
   if (the s looses useful FoV) {
      if (the lost FoV of s > p) {
         *search the first available camera in
            the redundancy group of s;
         if (the camera is available and is off) {
            *turn it on;
            *set the time when it was turned on;
            *turn s off;
            *set s' time when it was turned off;
         }
         else if (no camera was found)
                                    *keep s on;
```

```
13.         }
14.     }
15. }
```

8.3. Pseudo-code for choosing the camera that best recovers the lost FoV

This part is responsible for determining in a dynamic manner the best camera that can recover the lost FoV of a neighboring sensor. A sensor S that has its useful FoV obstructed in a percentage greater than a threshold p is not considered efficient anymore. Therefore the best available camera that can recover its FoV is searched in its redundancy group. Sometimes, a camera might not be available due to communication problems. This aspect will be explained further in the chapter. The camera that recovers the FoV has to be off. This is an important condition because the consequence is that for every turned off camera, another camera that recovers its FoV is turned on, so the coverage is maintained at a certain level. When no available camera is found in the redundancy group, the sensor is maintained on even though for the obstruction period its effectiveness is very low. In Figure 8.3. at lines (7) and (9) different times are set. This is necessary because the purpose is to maintain the coverage as high as possible. It is important to mention that in the beginning, after all the sensors compute their redundancy groups, only one sensor from each redundancy group remains on the rest being turned off. This measure is taken not to overload the communication bandwidth and also to prolong the lifetime of the network. From each redundancy group only the sensor that has the greatest useful FoV remains on. This situation is considered to be the optimal one. After the obstacle that obstructed the FoV of a sensor S has passed we want to go back to the optimal configuration. Therefore, if no other obstacle appears after one step S will be turn on again. This way, the algorithm will always try to reach the optimal sensor configuration and preserves coverage in a dynamic way with respect to the unpredictable traffic situations. In this implementation we improved the way the algorithm reaches the optimum sensor configuration. Indeed, in the previous version of it we only turned on the sensor that was turned off due to obstruction if the total coverage obtained this way was greater than the present one.

8.4.2. Algorithm With Prediction

The algorithm that implements the concept of prediction also uses redundancy. In this case redundancy is used to avoid the case in which two or more redundant sensors are on, in the same time. The difference is the fact that if a sensor is turned off, the computation of another sensor to be turned on is done at the network level.

First all the sensors are ordered with respect to the useful FoV determined by the intersection between the FoV of the sensor and the road. The sensors are then ordered ascending from the sensor with the least coverage. Next, the redundancy between sensors is computed keeping on the sensors with better coverage.

```
for (*all the sensors in the network){
    *determine the FoV intersection with the road;
    *sort the sensors ascending according to their coverage
    }
for (*each camera s in the descending order){
    *compute s' redundant cameras from its left, by
   computing common road coverage with every other sensors;
    }
```

8.4. Pseudo-code for ordering and computing the redundancy

K represents the percentage of common useful intersection between the FoV of two cameras in order for the cameras to be considered redundant. If the useful common area of two sensors is greater than k, the sensors are considered redundant.

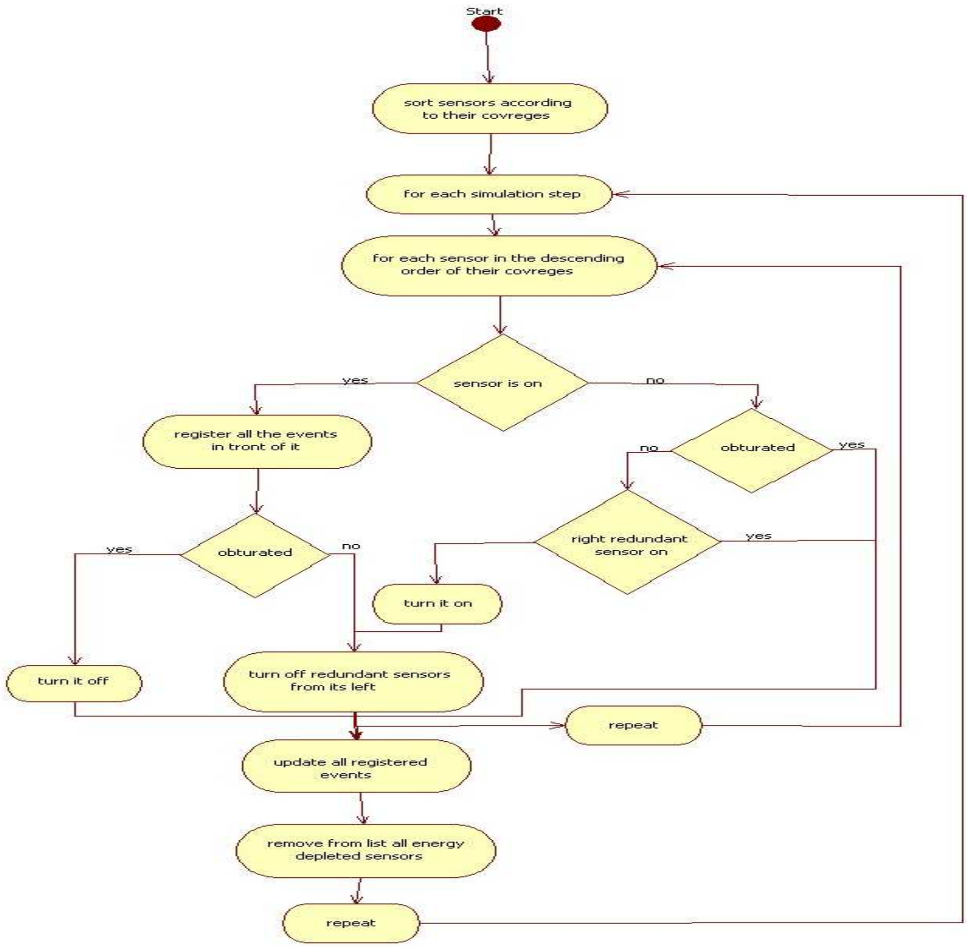
Below, in Figure 8.5. is described the process that does the management of the sensors turning them on and off.

The idea for choosing the best sensor is to compute for all the sensors their intersection with the road and according to their useful FoV to order them ascending. The next step is to turn on all the sensors. Then traverse the list of ordered sensors from the end to head and turn off the sensors that are redundant with it. In the end the sensors that have the best coverage and are not redundant with each other will be on. The rest would be turned off. This is the starting situation of the sensors. If an event enters the FoV of a sensor and obturates it, another sensor that is off and offers the best coverage among the turned off sensors is searched and if it is not redundant with any already turned on sensor, it will be turned on. The obturated sensor will be then turned off. It will turn itself on again when the event would have passed its FoV according to the prediction. If the event has not passed yet, the synchronization of the real position of that event with the events database will be realized.

```
*turn on all the sensors
*at each simulation step{
    *the ascending list of sensors is traversed from the end
        to the head{
            if(sensor is on){
                *register all events in front of it
                if(obturated()){
                    *turn it off
                    }
                }else{
                if(not obturated()){
```

```
                    if(not right redundant sensor on){
                        *turn it on
                    }
                }
            }
        if(sensor is on){
            *turn off its redundant sensors from its
               left
        }
    *update all registered events
    *remove from list all energy depleted sensors
    }
```

8.5. Pseudo-code for choosing the camera that best recovers the lost FoV



8.6. Activity diagram for choosing the camera that best recovers the lost FoV

Figure 8.5. represents the pseudo code for choosing the best camera that recovers the FoV of the obstructed sensors. The case weather there still is energy left in the next best sensor that potentially would be chosen for replacement is also treated.

Figure 8.6. shows the activity diagram corresponding to the pseudo code presented above.

8.5. Mathematical Model for Monte Carlo Simulation

The mathematical model that is presented below computes the degree of road occupation with cars and how this degree is influenced by the vehicles' dynamic data. This is important because the number of vehicles influence the degree of sensor obturation.

We consider *R* = the degree of road fullness with cars (%)

So = the length of the occupied road segment (m)

Ls = the length of the road (m)

$$R = \frac{So}{Ls} \tag{84}$$

The mathematical model considers two cases:

(a) road saturated with vehicles and

(b) road unsaturated with vehicles.

Saturation condition: $\forall\ c_i \in Cars, dist(c_i, succ(c_i)) \leq dist_{min}(c_i)$ (85)

Insaturation condition: $\exists\ c \in Cars, dist\big(c, succ(c)\big) > dist_{min}(c)$ (86)

Where $succ(c)$ is the successor of *c* and $dist_{min}(c)$ is the minimum distance at which *c* can efficiently respond at any change of the traffic conditions that arise in front of *c*. $dist_{min}(c)$ depends on a lot of factors like speed, technical status of the vehicle, the degree of car charging weight, the road geometry, the status of the road and the driver's driving skills. Technically,

$$dist_{min}(c) = minimum\ distance\ that\ provides\ enough\ time\ to\ react\ to\ the\ traffic\ dynamics$$

If we consider ≅ 1 second the analysis and biological reaction time and another 1 second the mechanical response of the car to the driver's command, we obtain $dist_{min}(c)$ = the distance that is made in 2 seconds:

$$dist_{min}(c) \cong v_c * 2 \tag{87}$$

Where v_c is the speed of the car (m/s).

At saturation:

$$Ls = \sum (L_{C_i} + dist_{min}(c_i)) \tag{88}$$

Where L_{C_i} is the length of c_i (m)

$$Rs = \frac{\sum L_{C_i}}{\sum (L_{C_i} + dist_{min}(c_i))} = \frac{\sum L_{C_i}}{\sum (L_{C_i} + 2v_{C_i)}} \tag{89}$$

Observation: R at saturation has a maximum value.

The lengths of the cars are given by a probability function, $P_{L_{c_i}}(l)$, the probability that the car c_i has the length l, l∈ $[l_{min,}l_{max}]$.

$$\int_l P_{L_{c_i}}(l) = 1 \tag{90}$$

due to the fact that the probability integral is computed on the entire probability domain. $P_{L_c}(l)$ is obtained from the real collected data set and can vary with respect to the time of day (e.g. at a certain hour, there may me more trucks than cars) and to the lane (e.g. on the speed lanes, are more cars than trucks). Considering the two variables, we detail (90): $P'_{L_c}(l, t, w)$, where t is the time of day and w is the lane number.

Consider $$L_c(p, t, w) = P'^{-1}_{L_c}(l) \tag{91}$$

where L_c is a function that represents the length of a car with respect to the probability of that length to be generated on the lane and $P'^{-1}_{L_c}(l)$ is the reverse after l of P'_{L_c} , p has the probability =1

=> $$\sum L_{c_i} = \int_{p\in[0,1]} L_c(p, t, w) \tag{92}$$

Analogous with the lengths, the speed of the cars are also given by a probability function, $P_{v_c}(\mathrm{v})$ is obtained from the real collected data set and can vary with respect to the time of day (e.g. at rush hour, the speed is slower) and to the lane (e.g. on the first lane, the speed range is slower than on the second lane called the speed lane).

$$\int_v P_{v_{c_i}}(\mathrm{v}) = 1 \tag{93}$$

Considering the two variables, we detail equation (90): $P'_{v_{c_i}}(\mathrm{v}, \mathrm{t}, \mathrm{w})$.

Consider $$v_c(p,t,w) = P'^{-1}_{v_c}(v) \qquad (94)$$

where v_c is a function that represents the speed of a car with respect to the probability of that speed to be generated on the lane and $P'^{-1}_{v_c}(v)$ is the reverse after *v* of $P'_{v_c}(\mathrm{v}, \mathrm{t}, \mathrm{w})$, has the probability =1

=> $$\sum v_{c_i} = \int_{p\in[0,1]} v_c(p,t,w) \qquad (95)$$

Equation (89) becomes:

$$Rs = \frac{\int_p L_c(p,t,w)}{\int_p L_c(p,t,w) + 2\int_p v_c(p,t,w)} \qquad (96)$$

where $p \in [0,1]$

At road insaturation, statistically, we can use a density function for the cars: $N_c(l)$ that represents the number of cars with length l that exist on the road. This function can also vary with respect to the time of day (e.g. at a certain hour, there may me more trucks than cars) and to the lane (e.g. on the speed lanes, are more cars than trucks). Considering these two variables, we introduce the detaliation of $N_c(l)$ as $N'_c(l,t,w)$. We compose $N'_c(l,t,w)$ with $L_c(p,t,w)$ in order to obtain the total loadness of the road:

$$R_{Ns} = \frac{\sum So}{Ls} = \frac{\int_p (N'_c(L_c(p,t,w),t,w) L_c(p,t,w))}{Ls} \qquad (97)$$

where $p \in [0,1]$

Equation (97) represents the loadness of the road when it is not saturated.

For a better understanding, we can consider a simple example; *p*=0.3 and for *p*=0.3 we can consider $L_c(0.3, t_0, w_0)$=4 m and $N'_c(4, t_0, w_0)$ = 20 cars => the surface of the road occupied by cars that have a probability *p*=0.3 is $So_{p=0.3}$ = 4*20 =80 m.

8.6. Monte Carlo Simulation Description

We used Monte Carlo to simulate a real situation and to test the performance of the proposed algorithm. The author of [109] offers a study regarding the distribution of types of cars (vehicles and trucks) on lanes together with their speeds during 24h. The differentiation of vehicles first is done only in two categories – cars and trucks. The study than gives a more detailed approximation regarding the lengths of the cars and the lengths of the trucks, also on lanes and on a day duration. For the simulation, those data were collected, processed and used for providing a real situation for simulating the Monte Carlo method. For correct data processing, [110] and [111] were also used.

The performed simulation was made on a road with 2 lanes per direction. The factors considered are the types of the vehicles (cars/trucks), the number of the lanes with the remark that trucks have a much better probability to be on the first lane and cars have a much better probability to be on the speed lane, according to collected data. Vehicles have the possibility to overtake other vehicles on the lanes. The speeds and the distribution on lanes are faithful to the collected data only for the initial simulation step on lanes. When a vehicle enters the road that corresponds to the probability of the collected data that closely follows the real scenario, it has a certain speed, also corresponding to the real scenario. We called this speed *prefferedSpeed*, in order to be able to simulate closely the real traffic scenario. So, if we presume that we have vehicle M1 on lane 1 driving with the speed v1 (*prefferedSpeed*) and at some point it gets closely to another vehicle M2, driving with the speed v2 (*prefferedSpeed*), v2<v1, M1 will try to overtake M2. In order to do this, M1 checks to see the distance between its current position and the closest car behind and before it on lane 2. If the distance is sufficient it changes the lane and continues to drive with its *prefferedSpeed* until the first lane is free again. If the second lane is not free, M1 decelerates, and drives with the speed of M2 until it can overtake M2 and continues driving with its desired speed. Another remark is that a car cannot get closer to another car that is under certain limit computed from the driving manual and that gives enough time to observe an unpredicted event and to react to that event. The pseudo-code is presented in Figure 5.7

There is a tiny difference from changing the lane from lane1 to lane2 and changing the lane from lane2 to lane1. A vehicle can change the speed lane (lane2) to the first lane only if it doesn't have a car in front that has a slower speed. The purpose of this condition is to avoid overtaking on the right side of the car. This situation is solved by the fact that all cars from the speed lane constantly try to come back to the first lane if the condition regarding the distance between cars allows it.

The cars represent the dynamic obstacles that interfere in the useful FoV of the sensors and opturates the sensors, point in which the other algorithm for coverage preservation does its job.

```
if(M1 is on lane1){
     if(the prefferedSpeed of M1< prefferedSpeed of M2) {
         if(the distance between M1 and M2 > minDistBetweenCars) {
                *check if lane2 is free;
                   if (lane2 is free)
                      *M1 changes the lane to lane2;
                      }
                   else {

     *M1 decelerates until it has the speed of M2;
```

```
*M1 gets closer to M2 until distBetweenCars =
 minDistBetweenCars;
*when lane2 is free, M1 changes the lane to
 lane2;
if(M1 changed the lane to lane2)
     *the speed of M1 = its preferredSpeed;
     }
}
```

8.7. Pseudo-code for changing from lane 1 to lane 2

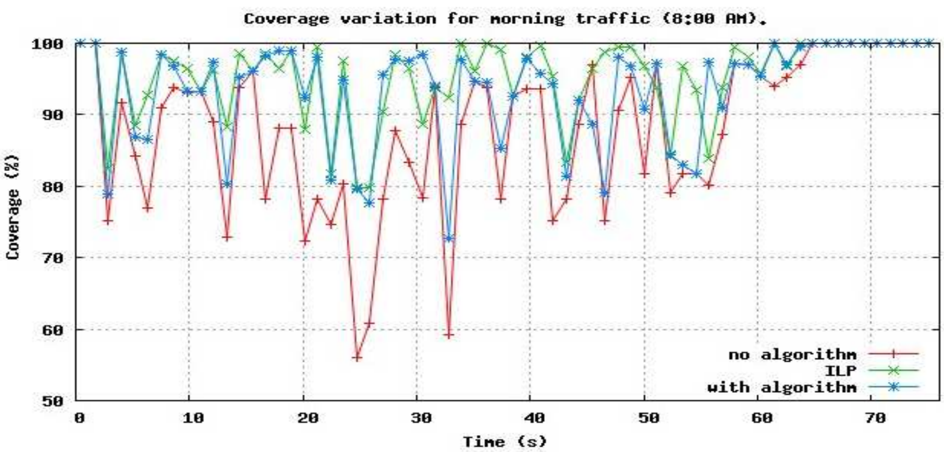
8.7. Monte Carlo Simulation

The experimental process exposed in this chapter regards several platforms and algorithms that are combined and compared. The new version of the algorithm was implemented as a Java program and run on a home desktop. This version of the algorithm is composed by three distinct algorithms: the one presented in Chapter 3, the implementation of another algorithm that computes the FoV loss, without taking any action regarding sensor management, and the LP-solve algorithm. The latter was developed also in Java. We have modeled the algorithm by writing equations for each constraint, for example from each redundancy group, only one sensor remains on, or another example can be the restriction that for each sensor that is turned off due to obstruction, another sensor is turned on, instead, etc and by maximizing the coverage equation. These equations represent the mathematical model of the algorithm and were solved by LP-solve, that was included in our Java program.

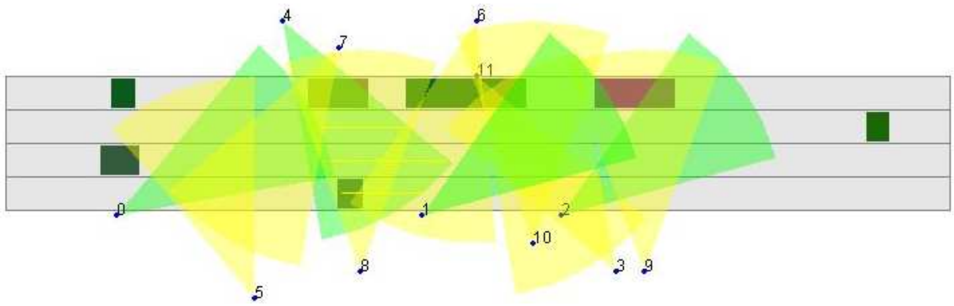
Monte Carlo simulation was also developed as a Java program and run together with the algorithms above on the same home desktop PC. MC simulation was developed from its mathematical model, presented in Chapter 8.4. and was the basis for all the algorithms because it provided them real test cases and real traffic behavior. The data that were collected from reality were processed and then represented as distributions in an .xml file. We have chosen representing the data as distributions due to the large quantity of data collected in a day's time. There were a lot of tests performed. For example, one test was to run MC simulation for one day, in order to get through all densities and to detect all possible anomalies that can appear. The tests were performed for a road for which data traffic was gathered in studies [109, 110]. The resulted file is huge because the data regarding coverage was collected every 0.5 second. It was time demanding to analyze the resulting file, and for space reasons we will present only the most representative snapshots from that file. In order to be convinced by the validity of the results, considerable many parts of the simulation were run up to 3 times more. The simulation that lasted 1 day was performed for the proposed improved algorithm. Due to scalability reasons of LP-solve, this algorithm was run only for the snapshots mentioned above in order to compare the efficiency of the proposed algorithm for coverage preservation in the presence of dynamic disturbance.

Figure 10.8. represents a simulation portion from the 1 day simulation, run separately, but with the same data corresponding to the exact period of day. It can be observed that the proposed algorithm is close to ILP algorithm as performance. ILP is the best because it mathematically computes which sensor to turn on in order to have the best overall coverage. The major disadvantage of ILP is its scalability. The algorithm was modeled with ILP, as well, to prove the dependability of our algorithm. The traffic distribution has a Gauss curve with its peak at lunch break hour, but close also at 16:00 hour. It is important to observe that if no sensor management is performed, the difference can be up to 20%.

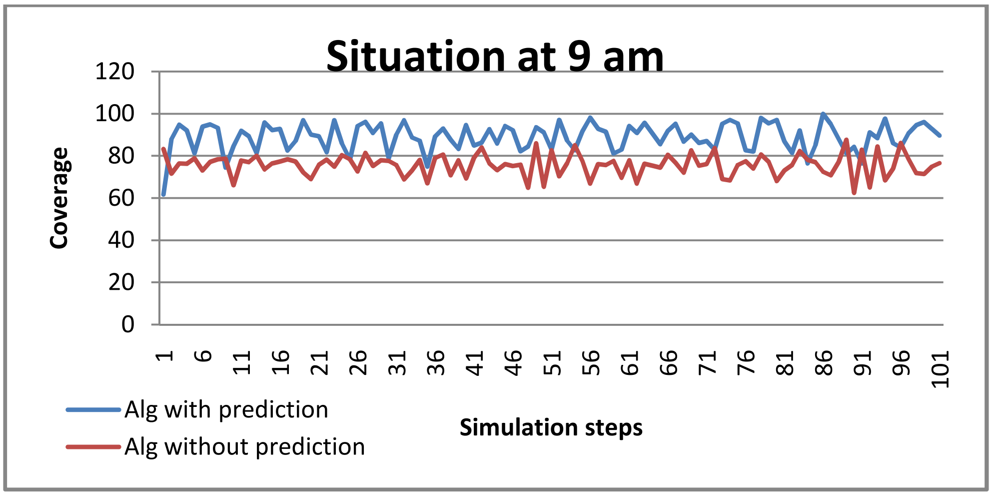
Figure 10.9. shows a frozen simulation step from 8:00 a.m. In the presented tests, the configuration of the sensors remains the same, 12 sensors arranged like in Figure 8.9., but only 4 active at a time. The number of active sensors at a time is established by the condition that from each redundancy group, only the sensor that has the best coverage is kept on. The active sensors change at each simulation step, as described in Chapter 8. MC analysis results showed that the average number of cars at 12 o'clock present on the road varied between 6-9.



8.8. Coverage preservation for traffic monitoring at 8:00 a.m.



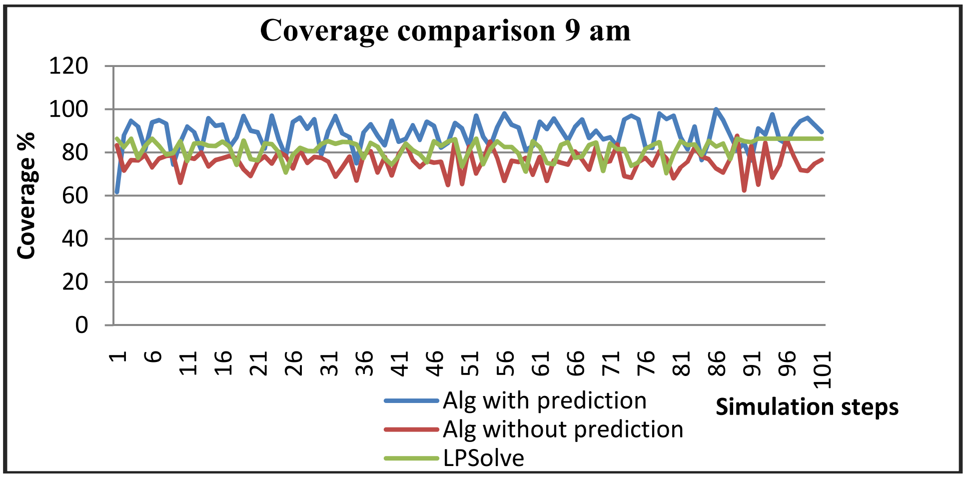
8.9. Road situation at 8:00



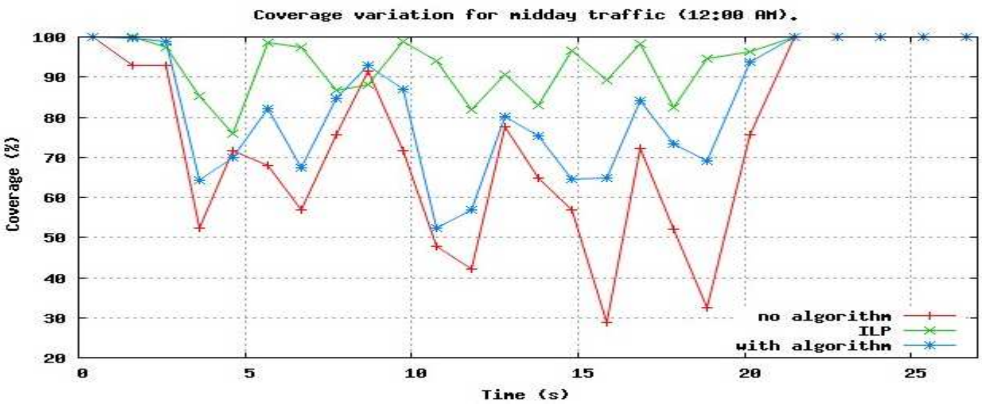
8.10. Road situation at 9:00 with and without prediction

Figure 8.10. represents the coverage determined with the first algorithm, without prediction ad with the prediction algorithm at 9 am. The traffic is higher than at 8 am due the fact that in most of the places, work starts at 9am. The differences between the algorithms come from the fact that in the case of the algorithm without prediction the best sensor to replace the obstructed one is chosen from the redundancy group. This corresponds to finding a maximum in a local. In the case of the algorithm with prediction the best sensor to recover the coverage is searched at the network level. Both algorithms perform well, but the algorithm with prediction performs even better than the other one.

For all the encountered situations used for simulation, between the minimum number of cars and jam situation, the results of the coverage variation for the proposed algorithm without prediction was kept around 7% with respect to ILP, which is considered optimal from mathematical point of view.



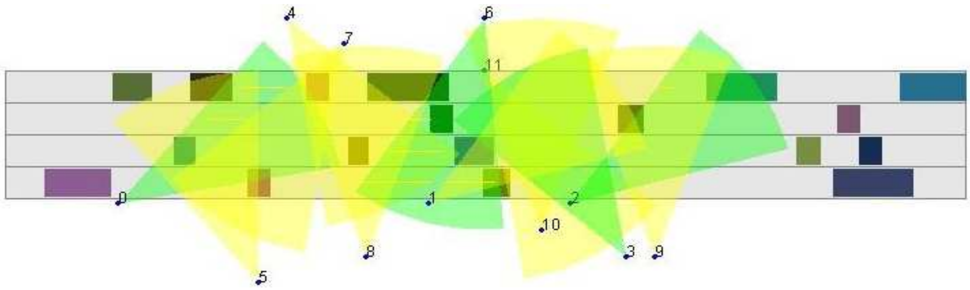
8.11. Coverage variation for traffic monitoring at midday



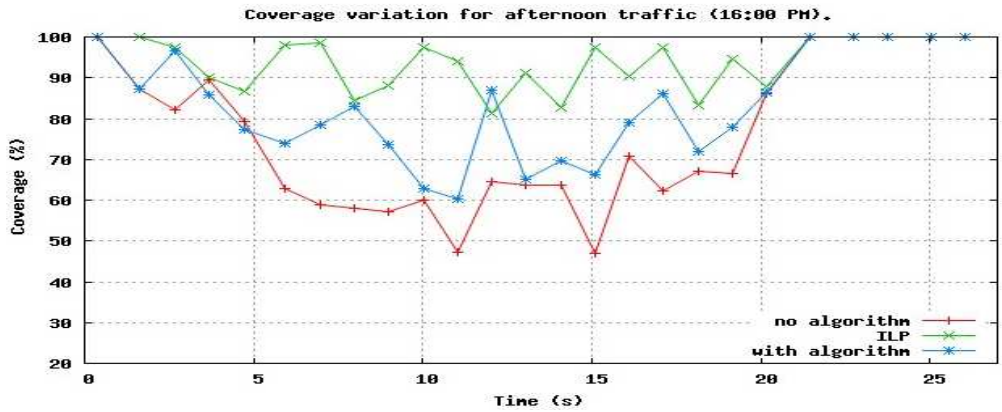
8.12. Coverage variation for traffic monitoring at midday

In Figure 8.11. is the situation presented also in Figure 8.10. with the difference that we showed the performance of the of the prediction algorithm also compared with LPSolve. It can be seen that the prediction algorithm is better even then LPSolve that mathematically computes the best sensor to be turned on. Still, as explained above, LPSolve, as well as the algorithm without prediction uses the concept of redundancy groups of the obstructed sensor to choose another sensor for replacement.

Figure 8.12. represents the coverage preservation situation at midday. The remark is that for midday (Figure 8.12.) and for 16:00 o'clock, we had to enlarge the interval in which we collected data at 1.5 seconds from 0.5 seconds due to ILP scallation. In these cases the number of cars that are on the road is considerably higher and the computation was slower. The test scenario remained the same. Figure 8.13. shows a picture from road midday situation. It is obvious that the number of cars has a great impact on coverage. This case scenario is taken during the lunch break. While the case where no algorithm management is applied, the coverage dropped under 30%, while the proposed algorithm kept the coverage level above 50%. If no algorithm is applied, no sensors will turn on to replace the obstructed ones.



8.13. Road situation at 12:00



8.14. Coverage variation for traffic monitoring at 4 pm

In order to prove the dependability of the proposed algorithm, we ran different test scenarios in which different sensor arrangement was considered. The experiments section presented the worst case found. At different simulation sensor arrangement scenarios, the presented case remained the one presented in Figure 8.14.

8.8. Conclusion

This chapter proposes to prove the dependability of real-time dynamic coverage preservation algorithm considering the presence of undeterministic disturbances using MC simulation. The goal is to minimize the FOV loss of the cameras due to dynamic obstacles by identifying the best set of additional cameras that can compensate for the loss. The set is identified under the timing and sampling constraints of the application and with the objective to reduce the utilized resources. As wireless communication can be unreliable, the availability of a camera is also considered. The dependability of the algorithm was proven by the reliable tests that were made in which the proposed algorithm was compared with its ILP model that mathematically determined the best solution for coverage preservation. The proposed method was simulated using the MC technique that showed that in the worst case, almost at saturation, the coverage is maintained above 50% with the minimum of resources used.

The algorithm is fast and scales well for large number of cameras and monitored vehicles. It is useful for reliable data acquisition over extended periods of time, including video image collection for traffic monitoring applications. This chapter proved its dependability.

The performance of the prediction algorithm was also presented and compared with the algorithm mentioned above as well as with LPSolve. The performance of the algorithm with prediction taking coverage as the metric is the best.

9. Coverage Variance When Sensors Have Limited Energy

9.1. Abstract

The performances of the algorithm with prediction and of the algorithm without prediction are presented in this chapter. The sensors can be divided into two categories: with and without rechargeable batteries. The performance of the algorithms that have unlimited amount of energy is presented in the above chapters. The performance of the sensors that have limited amount of energy is presented in this chapter. The algorithms perform well from both the coverage perspective as well as from the lifetime of the network, as seen in Figure 9.1.

The performance of the algorithm with prediction is better in the first part, but the algorithm with redundancy groups is better in the second part after sensors start to remain without battery. The reason for this is that the algorithm with prediction can use more sensors at a time between which redundancy is 0 and for a turned off sensor more sensors can be turned on at the network level while the algorithm without prediction uses redundancy groups and the algorithm turns on sensor from the redundancy group of the obstructed sensor. As a result the distribution of the turned on sensors can be better at this algorithm.

Both algorithms perform well but each has its advantages and particularities. Choosing one instead of the other one depends on the purpose of the application.

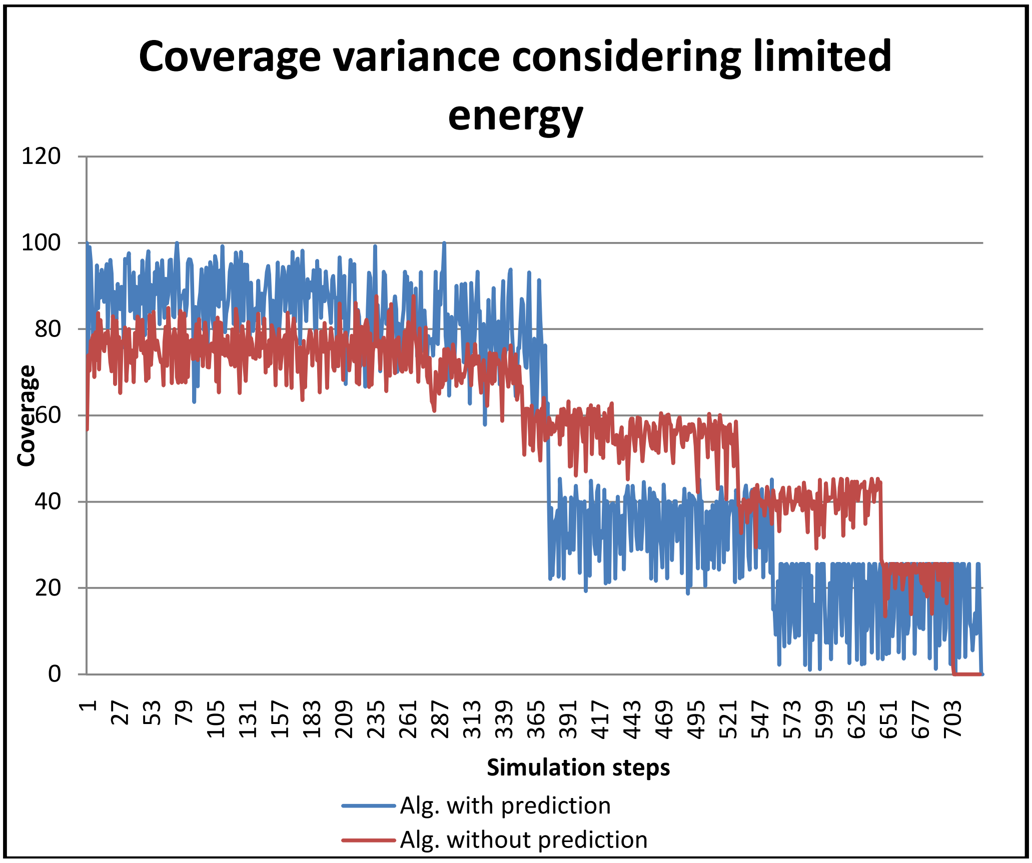
9.2. Coverage Variance

Wireless sensors can be of two types: they can have rechargeable batteries or they can have batteries that die after a while.

The performance of the proposed algorithms was realized in the above chapters for sensors that have unlimited amount of energy.

In this chapter we present the performance of the proposed algorithms: the one using redundancy groups and the prediction one from the energy perspective, showing how these algorithms are capable to offer good coverage performance while also prolonging the lifetime of the network.

For the simulation performed no particular type of sensor was used due to the fact that the battery chosen for the network can vary. The idea used was to establish an amount of time (that in our simulator can set) that represents the time the sensors can be on. After this time, the battery of the sensors is considered to be finished.



9.1. Coverage variance considering limited energy

Figure 9.1. presents the coverage variance for the two algorithms if the sensors have limited amount of battery.

The results are interesting. In the tests performed above (see Figures 7.1., 7.2., 8.10., 8.11.), the prediction algorithm always performed better than the algorithm without prediction. In Figure 9.1 it can be observed that in the first part, the prediction algorithm performs better, but in the second half, the algorithm using redundancy groups performs better. The explanation is the fact that in the first part, for both algorithms, sensors have energy and the performance of the prediction algorithm is better, as shown in previous chapters. As presented at the algorithms description at both algorithms if they have energy resources, the best sensors are used, so the algorithm performance is their best. After energy starts to finish those sensors are replaced by other sensors that do not perform as well as did the ones that consumed their battery. The performance of the algorithm without prediction is better in the second part due to the fact that the algorithm tries to turn on sensors from the redundancy group of the sensor that drained its battery, so the algorithm tries to turn on sensors from each redundancy group and as a result the algorithm offers as good coverage as possible along the entire road. The second algorithm with prediction performs better in the first part where the energy is not an issue, but after the energy starts to finish, the coverage recovery is weaker due to the fact that more sensors are used at a time, sensors that give a better coverage in the first part.

It is obvious in Figure 6.1 that the decrease is in steps. The steps come from the fact that at each moment the best sensors are used and when they become unuseful, the network tries to replace them, but the only sensors that still have energy are the less performant ones.

9.3. Conclusion

This chapter presents the performance of the proposed algorithms for the sensors have limited amount of energy. The algorithms perform well and by using only certain sensors at the time, the lifetime of the network is prolonged. If there would be no algorithm for sensor management applied, the network would have lived until the first decreasing step, around simulation step 360. Each of the algorithms has advantages and disadvantages. Choosing the algorithm for the network depend on the purpose of the application. If the resources are unlimited, the algorithm with prediction performs better, but if the resources are limited, the overall coverage of the algorithm without prediction that uses redundancy groups is better.

10. TRAFFIC BEHAVIOR SIMULATOR *SIMULO*

10.1. Abstract

This chapter begins by presenting the traffic behavior aspects that are implemented and also shows the impact that several variations have upon coverage. For example the number of speedy cars and the value of the minimum distance between cars were varied in order to determine the influence upon coverage.

A thorough study about the existing simulators is then described and the motivation for developing a new simulator is explained. Next, the description of our simulator, Simulo is presented. Simulo integrates both micro and macro simulation. Besides this, Simulo closely simulates the real traffic rules that imply overtaking, acceleration, deceleration, coming on the first lane if it is free, etc. All these rules are applied depending on the type of human driving, presented in the next subchapter. The mathematical background for the simulator is the one presented in Chapter 8. Conclusions are drawn at the end.

10.2. Traffic Behavior

This chapter presents the influences of the driving manner upon coverage determined by both algorithms. In traffic drivers have behave differently. In our work we divided the behavior into normal behavior and speedy behavior. The simulator acts different if the vehicles' behavior is speedy. The differences are first of all the speed that is significantly higher than the normal average speed. Also besides the speed, if the simulator determines that a vehicle is speedy, it allows that vehicle to overtake on the right side, if a minimum safe distance between cars allows it. Moreover, for speedy vehicles, the simulator also adjusts the minimum distance between cars by reducing it with 20% from the normal distance between cars applied at normal behavior.

The simulation driving behavior capabilities will be presented in the final thesis. Still, to make an idea, the simulator follows pretty closely the real behavior. Each vehicle enters the road with a certain speed on a certain lane. That speed is called preferred speed. If the vehicle is on the second lane and the first lane is free, it automatically switches lanes. If a vehicle has a car in front that is slower, it decreases its speed and drives at the safe distance from the car in front. If it can overtake the car, it does that and then if possible comes back on the first lane, trying to accelerate until it reaches its preferred speed.

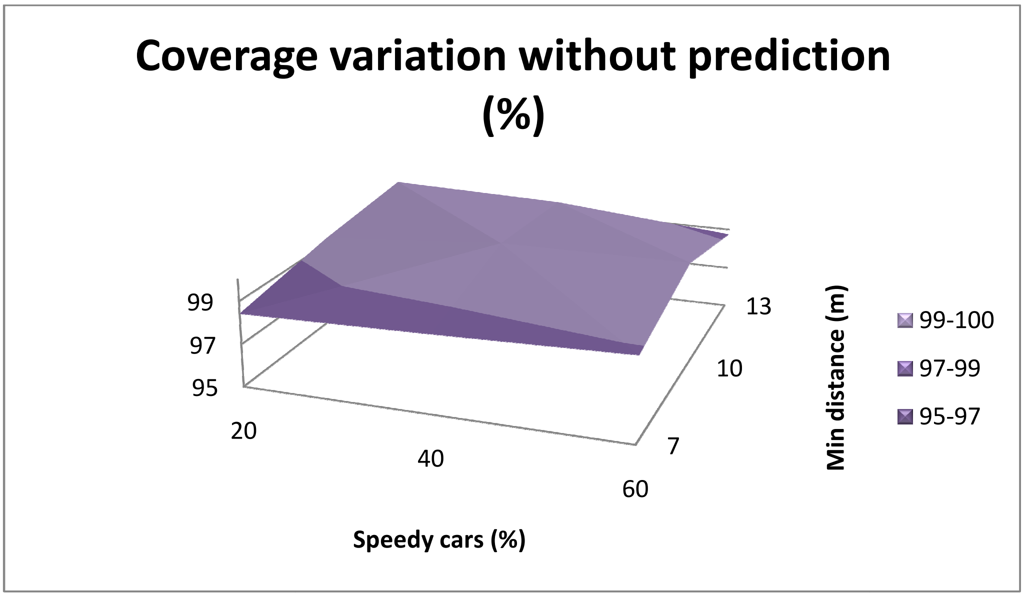
Vehicles are also divided into cars and trucks. Trucks are not allowed to exceed a certain speed limit.

When tested on different traffic behavior situations, both algorithms (with and without prediction) show very good stability. The test cases included variations of percentage of speedy cars, between 20–60% and of the minimum distance between cars 7-13 m. For testing we used the case with limited amount of energy for the sensors.

The metric used was total coverage (TC):

where :

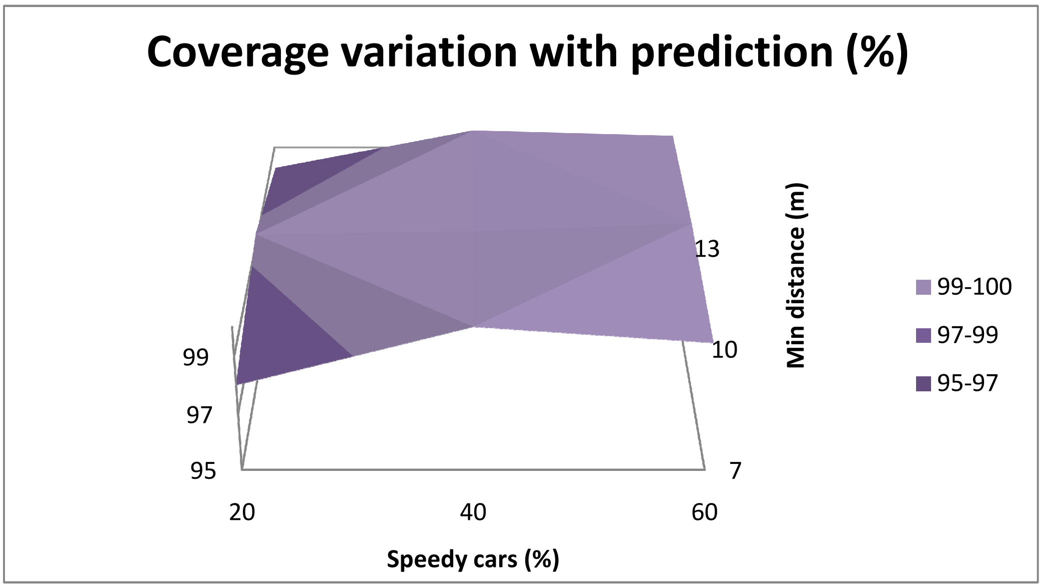
- TC – total coverage
- C(t) – coverage at moment t
- t_0 – simulation starting time
- t_f – simulation ending time, when no energy is left at any sensor



10.1. Coverage variation for algorithm without prediction

The results show a maximum variation of about 2% for both algorithms. There is no noticeable difference between the performances of the algorithms.

The variation is more dependent of the parentage of speedy cars than of the minimum distance between cars. The cause is related to the acquisition and processing time of the sensors, so as the car speed increases, the senor analysis quality decreases.



10.2. Coverage variation for algorithm with prediction

10.3. Inroduction

Vehicular Ad-Hoc Networks (VANET) represent a growing area of interest due to the large number of vehicles on the roads. Traffic rises many issues due to jams, time spend in traffic, pollution, accidents, etc. The studies regarding traffic flow are of much need in order to provide more safety on the roads, to reduce the number of accidents by introducing or synchronizing traffic lights. The implementation of such a system and more over adjusting its parameters until it reaches its goal can be really expensive. That is the main reason for simulating traffic.

There is quite number of traffic simulators specific for the purpose of their design: simulators for urban traffic, simulators for highways, simulators for car crashes. Each simulator has certain specific facilities. From one perspective they can be divided into two categories: simulators for micro traffic or simulators for macro traffic. Micro traffic simulation refers to simulation at the level of vehicles, while macro traffic simulation refers to the flow of the vehicles. One example of micro traffic simulation is presented [112]. There are simulators that combine the two of them like VanetMobiSim [113]. The idea used for this simulation is to put sensors on vehicles and they can provide at each time both micro and macro traffic information. The drawback is that this is performed dynamically, so the information are from different parts of the road depending on the position of the vehicle. VanetMobiSim simulator uses graphs in order to simulate traffic and does not take into account the vehicles' length. Also, in this simulator human behavior is not at all modeled. The author of [114] presents a study regarding human driving behavior. Mainly it studies the reaction time of the driver if different factors interfere in the usual traffic flow.

	VanetMobiSim	SUMO	MOVE	STRAW	FreeSim	CityMob
Software						
Portability	✓	✓	✓	✓	✓	✓
Freeware	✓	✓	✓	✓	✓	✓
Opensource	✓	✓	✓	✓	✓	✓
Console	×	✓	✓	—	×	✓
GUI	✓	✓	✓	✓	✓	✓
Available examples	✓	✓	✓	—	✓	×
Continuous development	×	✓	×	×	—	✓
Ease of setup	Moderate	Moderate	Easy	Moderate	Easy	Easy
Ease of use	Moderate	Hard	Moderate	Moderate	Easy	Easy
Maps						
Real	✓	✓	✓	✓	✓	×
User defined	✓	✓	✓	—	×	×
Random	✓	✓	✓	×	×	✓
Manhattan	×	×	×	×	×	✓
Voronoi	✓	×	×	×	×	×
Mobility						
Random waypoint	✓	✓	✓	×	×	✓
STRAW	×	✓	✓	✓	×	×
Manhattan	×	✓	✓	×	×	✓
Downtown	×	×	×	×	×	✓
Traffic models						
Macroscopic	×	×	×	×	✓	×
Microscopic	✓	✓	✓	✓	✓	✓
Multilane roads	✓	✓	✓	✓	—	✓
Lane changing	✓	✓	✓	✓	—	✓
Separate directional flows	✓	✓	✓	✓	—	✓
Speed constraints	✓	✓	✓	✓	✓	✓
Traffic signs	✓	✓	✓	✓	—	✓
Intersections management	✓	✓	✓	—	—	×
Overtaking criteria	✓	—	—	—	—	×
Large road networks	—	✓	✓	✓	—	✓
Collision free movement	—	✓	✓	—	—	✓
Different vehicle types	×	✓	✓	—	×	✓
Hierarchy of junction types	×	✓	✓	—	×	×
Route calculation	✓	✓	✓	✓	✓	×
Traces						
ns-2 trace support	✓	×	✓	×	×	✓
GloMoSim support	✓	×	✓	×	×	×
QualNet support	✓	×	✓	×	×	×
SWANS support	×	×	×	✓	×	×
XML-based trace support	✓	×	×	×	×	×
Import different formats	✓	✓	✓	×	×	×

Table 10.1 Comparison of mobility generators [117]

When talking about simulators, some general characteristics should be mentioned besides the degree of generalization they perform (micro/macro simulation).

- if the simulation is comprehensive or not- whether it simulates all traffic situations or not
- the type of analysis - what inputs they accept and how general or adapted to specific situations these data is
- what type of statistical data are generated
- what kind of decisions can be taken after analyzing the results
- the type of analysis the simulator performs
- what type of calibration has to be performed in order for the simulation to be performant
- if it allows intersection simulation
- if it allows multiple lanes on the same direction

- if it allows traffic lights, barriers
- if the simulator is capable of recording a simulation in order to repeat itif plug-ins are allowed in order to add new functionalities
- what is the output of the simulation: images, files, movies
- what type of license does the simulator have
- how it emulates the collection of real data
- can new sensors be added to the simulator
- what data can be varied in the simulator

Two most used simulators that are used for traffic simulation is VanetMobiSim [115] and SUMO [116]. SUMO is a well known simulator, but a major drawback is that it does not save the current traffic situation so that the exact simulation to be used again in order to perform comparisons between the performance of different algorithms. Also, human behavior is not modeled.

Moreover, as seen in Table 10.1, only FreeSim is capable of Lax macroscopic simulation, needed to simulate events, but its major drawbacks in comparison with Simulo are the fact that it does not support the simulation of different types of vehicles and it does not have multilane roads.

10.4. *Simulo* description

In this thesis we propose a new simulator, Simulo. This simulator has a mathematical foundation described in Chapter 10.4. The functionality of Simulo is presented next.

The configuration (the input) is loaded from a XML file. The simulation options are specified in this file and they regard:

- lanes with physical coordinations, lane id, neighbor lanes from right and left side if the case:

  <lane id='1' x0='80' y0='100' x1='350' y1='100' right='0' ms='70'>

Attribute	Description
id	the unique identifier of the lane
x0,y0	the starting point of the lane (m,m), considered at the middle of the lane's height. The direction of the cars on this lane are from its starting point to its end point.
x1,y1	the ending point of the lane
right	the id of another lane, if that lane is to the right of the current one, in the sense that a vehicle could change this lane to the one on the right
left	if there is another lane to the left
ms	maximum speed

Table 10.2 Lane attributes

- a lane contains one or more probabilistic vehicle distributions, grouped on types of vehicles (car, truck), each of them having the name of the type, the minimum and maximum length of the vehicle and the corresponding percentage of fleet:

  <cartype name='car' lmin='3' lmax='5' pof='98'>

  Each vehicle type has two probability distributions:
  - the density, that determines the number of vehicles that appear in the system starting with a specified hour:

    <stat name='density' type='linear' dataset='a' >

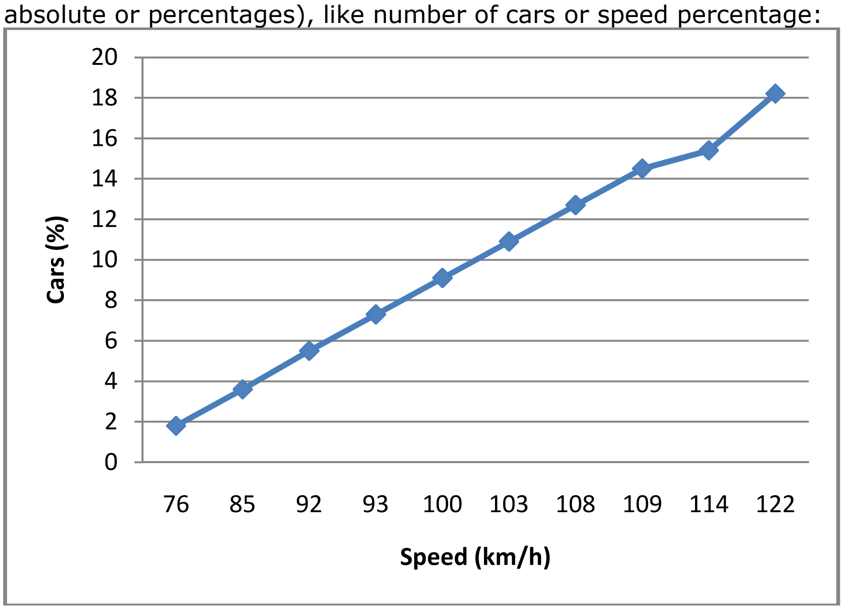
  - the velocity that determines the speeds of the vehicles that enter the system

    <stat name='speed' type='linear' dataset='p' >

Attribute	Description
name	the specific distribution
type	interpolation used between distribution's reference points
dataset	a=absolute values for the ordinate p=percentage values for the abscissa

Table 10.3 Vehicle attributes

Every statistic (stat) uses reference points, having on abscissa values like hour or speed and on ordinate a probabilistic value (in absolute or percentages), like number of cars or speed percentage:



10.3 Distribution example

According to the empirical data, collected by [109,110,111] there are mainly two types of distributions:

- linear - used for noncontinuous variations of the traffic flow, like: changing traffic lights, intersections, many people getting out of work, etc. All these are step variations in the density of the traffic flow, best approximated with linear interpolations between reference points.
- spline - used for continuous variations of the traffic flow, like: traffic variance between different hours when no external factors are interfering and there are no intersections or traffic lights, forming of car columns, etc

The probability distributions are conceived to be flexible and to accept more probability functions. Currently the multisegment distribution is implemented. More segments are specified and are interconnected in given points.

- for density, "x" represents the hour of the day and "y" represents the number of vehicles at that hour.
- for velocity, "x" represents the speed and "y" represents the percentage of vehicles that have that certain speed

- a sensor is defined by its coordinates, the bisector line with respect to the horizontal line, minimum and maximum communication and obstruction penalty. The wildness of the sensors is global and is given in the program:

<sensor x='100' y='90' a='30' minCommPen='0' maxCommPen='100' minObtPen='0' maxObtPen='80'/>

Attribute	Description
x,y	coordinate of the sensor (m,m)
a	angle of the sensor median with the world's abscissa
minCommPen, maxCommPen	minimum and maximum percentage of communication penalty
minObtPen, maxObtPen	minimum and maximum percentage of obturation penalty

Table 10.4 Sensor attributes

Due to the fact that other values like sensor type or wildness are the same for all sensors, they are given with the other global settings.

- other specifications that can be made in the XML file concern global settings:

<set lanewidth='6' newcarmintime='1' scenewidth='360' timeoffset='28800' speedycars='20' carmindist='10' alg="predict" />

Attribute	Description
lanewidth	width of a lane (m)

newcarmintime	minimum time before a new car is put on a lane (s)
scenewidth	width of the scene (m); the height is automatically computed to respect the proportion of the visualization window
timeoffset	the starting offset time of the simulation, from 0:00 (s)
speedycars	percent of speedy cars
carmindist	minimum distance between cars
alg	algorithm to be run: - lps - LPSolve - alg1 - algorithm without prediction - predict - algorithm with prediction

Table 10.5 Global attributes

- in order to be able to repeat a scenario, it can be specified if the cars can appear according to the probabilistic distributions or can be loaded from a file representing the data from a previous simulation.

```
<simulo>
<savedcars src='sc1013.csv'/>
<set lanewidth='6' newcarmintime='1' scenewidth='360' timeoffset='28800'  speedycars='0' carmindist='7' />

<lane id='1' x0='80' y0='100' x1='350' y1='100' right='0' ms='70'>
        <cartype name='car' lmin='3' lmax='5' pof='98'>
                <stat name='density' type='linear' dataset='a' >
                        <v x='0' y='350'/>
                        <v x='4' y='200'/>
                        <v x='8' y='750'/>
                        <v x='12' y='300'/>
                        <v x='16' y='400'/>
                        <v x='20' y='550'/>
                        <v x='24' y='350'/>
                        </stat>
                <stat name='speed' type='linear' dataset='p' >
                        <v x='76' y='1.8'/>
                        <v x='85' y='3.6'/>
                        <v x='92' y='5.5'/>
                        <v x='93' y='7.3'/>
                        <v x='100' y='9.1'/>
                        <v x='103' y='10.9'/>
                        <v x='108' y='12.7'/>
                        <v x='109' y='14.5'/>
                        <v x='114' y='16.4'/>
                        <v x='122' y='18.2'/>
                        </stat>
                </cartype>
        <cartype name='truck' lmin='6' lmax='15' pof='2'>
                <stat name='density' type='linear' dataset='a'>
                        <v x='0' y='10'/>
                        <v x='4' y='30'/>
                        <v x='8' y='100'/>
                        <v x='12' y='200'/>
                        <v x='16' y='150'/>
                        <v x='20' y='170'/>
                        <v x='24' y='10'/>
                        </stat>
                <stat name='speed' type='linear' dataset='p' >
                        <v x='84' y='1.8'/>
                        <v x='87' y='3.6'/>
                        <v x='90' y='5.5'/>
                        <v x='92' y='7.3'/>
                        <v x='93' y='9.1'/>
                        <v x='96' y='10.9'/>
                        <v x='100' y='12.7'/>
                        <v x='102' y='14.5'/>
                        <v x='105' y='16.4'/>
                        <v x='114' y='18.2'/>
                        </stat>
```

10.4 Simulo XML file example

At each simulation, the vehicles can be loaded from a presaved file, or they can be generated (according to the above probabilistic distribution) and in that case they are saved in a file for the possibility of being used in the future for diverse algorithms comparison using the same data sets.

All the coordinates are given in physical units that are then translated into screen coordinates.

In Figure 10.1. a XML file example is presented. On the second line it is shown that the simulation will load an existing file 1013, generated at a previous simulation.

- The simulation part consists of an object World that contains all the objects that intervene in simulation: vehicles, lanes, sensors, algorithms (derived from WorldObject)
- At each simulation step, for each WorldObject the methods *process* and *run* are called. They are responsible for the computation of the different aspects and for their display and are given as parameters the simulation current time, the time, the time interval from the last simulation step and the graphic drowing context. Each WorldObject also has an z-order, necessary for treating objects in a certain order. For example, the vehicles are processed after the lanes, so the lanes not to be drawn over the vehicles and hide them.
- In the simulator besides the physical objects there is an object *Alg*, that has different virtual aspects, for example the computation of the global situations that are related to the interaction between more objects and not a specific object (standby mode) and also related to the statistics
- The senors also have settings for minimum time intervals in which they can change their state (*minSwitchDeltaTime*). The sensors are displayed with green when they are on and with yellow when they are turned off, each sensor having specified its own id.
- The vehicles also have acceleration that can be negative, as well and respects the driving rules by the fact that they tend to always come back on the lane from the right. If a slower car appears in front, its velocity is tested to be slower and if the overtaking is possible. If it is not possible, the vehicle will reduce its own speed. In jam situation, the cars will keep a minimum distance between them. The lane changing cannot be performed too often, under a given interval. Each vehicle has a preferred speed and if from different reasons, it had to slow down, it will have the tendency to regain its preferred speed. The preferred speed is given by the distribution speed for its class of vehicles. For a nice visualization, the vehicles also have different colors, randomly chosen by the system and they also have displayed their id and speed.
- World also has methods for the management of simulation objects, for example a vehicle, after it exits the lane, it is erased from the simulation.
- The algorithms can change by changing a constant that specifies the current algorithm
- The mathematical part and the analytical - geometrical one is separately implemented and has functions for the computation of intersections, lengths, angles, coordinates, etc
- The statistical reporting part is conceived to save in a CSV format, for each set of values, being specified the current time and the value. The saving step is given by a constant and the system, when it saves the data for a

certain moment, computes between all the values from the last given interval and makes a range (the values are received at each simulation step, steps that can be more often than the step of results saving).

10.5. Conclusion

This chapter presented the description of Simulo, the proposed simulator used to determine the efficiency of the proposed algorithms. It is shown the need for developing and implementing a new simulation tool by a thorough study of the existing traffic simulation tools. Simulo has the possibility of simulating both micro and macro simulation, can save a simulation and reload the data to simulate the same traffic in order for the algorithm comparison to be accurate. The simulator has also the possibility of performing Monte Carlo simulation due to the fact that it loads XML files where data can be specified on hours and with respect to the probability of appearance for each event. Furthermore, Simulo is capable of simulating some human traffic behavior characteristics.

11. Conclusion. Contributions. Future Work

11.1 Conclusion

The thesis is divided into two parts. The first part refers to metrics used in wireless sensor networks. A detailed background of the metrics used in different types of applications that use WSN are described such as deployment metrics, coverage metrics, and energy saving metrics. The importance of metrics resides in the modality of determining the applicability area of the algorithms, in the comparison between algorithms, and in determining the resources necessary for the implementation of the algorithms. At the end of the first part, the proposed metrics are presented as a contribution. The proposed metrics correspond to two areas: metrics estimation for uncovered surfaces and paths and the second area Metrics for determining the influence of an algorithm for energy saving applied in the case of deployed and scattered sensors. Each of them has several metrics.

The second part refers to performance efficient algorithms for data collection in wireless sensor networks. It begins also with a solid state of the art regarding the algorithms used in WSN especially in target tracking. The issue raised is that at traffic surveillance vehicles can pass unnoticed if the sensors that are supposed to notice them are obstructed by other vehicles. The solutions proposed make use of a new concept, redundancy. Using redundancy two main algorithms were proposed for performant efficient data collection in wireless sensor networks. The management of the sensors used in the proposed algorithms turn sensors on and off, so the efficiency to be the high. Both algorithms have as metric coverage and also the prolonging the lifetime of the network. Both algorithms perform well for coverage as well as for prolonging the lifetime of the network.

One important aspect regarding the proposed algorithms is that the FoV recovery is performed dynamically. If a vehicle enters the FoV of a sensor and obstructs it, another sensor will be searched at the level of redundancy groups for the algorithm without prediction and at the network level for the algorithms with prediction. This way we manage to accomplish and maintain good coverage while keeping the minimum sensors on, still being the network able to realize a good coverage. From the algorithms perspective the minimum sensors that are on means the maximum number of sensors between which there is no redundancy. The goal is to minimize the FOV loss of the cameras due to dynamic obstacles by identifying the best set of additional cameras that can compensate for the loss. The set is identified under the timing and sampling constraints of the application and with the objective to reduce the utilized resources. The FOV loss changes dynamically depending on the traffic conditions. As wireless communication can be unreliable, the availability of a camera is also considered in the report. For the algorithm without prediction, two heuristic methods employ different cost functions for selecting the additional cameras used for FOV loss recovery were implemented. The cost functions are based on a new stochastic model for traffic monitoring, including the dynamics of mobile obstacles, unreliable communication, and resolution and timing constraints. The first cost function, Parameter Weighted Contribution (PWC), addresses deterministic situations by capturing the trade-off between the quality of recovery and the imposed timing constraints. PWC expresses the utility of a camera in FOV loss recovering, the available resources of a camera, and the capability of recovering multiple FOV losses of neighboring cameras. The second cost function, called Expected PWC (EPWC), addresses unreliable networks, such as wireless connections. EPWC extends PWC by incorporating the probability of a node to participate in FOV loss recovery, including the chances of the camera being obstructed by obstacles, experiencing data loss during communication, and other cameras used in the covering solution being also available.

In the case of the second algorithm the concept of events was introduced. An event is formed by several vehicles between which the distance is less or equal with the allowed minimum distance between vehicles, so no collision to appear. Events form and are split dynamically with respect to the dynamics of vehicles. In this algorithm another new concept is introduced. This algorithm makes use of prediction. We use prediction to determine the moment an event will pass the FoV of a certain sensor. This aspect is important because if a sensor that has the best coverage in its surrounding is obstructed by an event and turned off and after the event passes if this sensor is not turned on again, the loss recovery will be done by sensors less performant from the FoV perspective. In our case, each sensor can compute the moment the event is supposed to leave the FoV of the sensor and if the prediction is accurate, that sensor probably will turn on again in order to redo the coverage at the network level.

Prediction is a concept that also has the possibility of failing. The cases when the prediction is not accurate have several causes: at the beginning of the network, when the length of the events is not yet known or if an event splits or unites with another event. In these cases the prediction fails. It was shown that the prediction varies between 0 and 4%, which is an insignificant percentage.

As mentioned above, an event is formed by consecutive vehicles that are on the same lane, drive in the same direction and the distance between them is less or equal to the minimum distance allowed for collision free. If an event is formed, it will be seen by the sensors as a hole obstacle that obturates its FoV. It was also shown in the experiments that that the useful FoV representing the minimum distance between vehicles is negligible.

The performance of the algorithms shows that both algorithms perform well. The coverage is maintained almost permanently over 80%. They were tested using Monte Carlo simulation. Real data sets were used in order to obtain as close to reality results as possible. The data sets used were taken from a study for roads paving and it implied the division of vehicles into cars or trucks, their distribution on lanes as well as their speeds at different hours.

The mathematical model for traffic behavior was. The tests were performed taking into consideration multiple variations such as factors of human behavior presented in the last chapter.

The algorithms have a solid mathematical background and they were validated by articles, journal and also by tests performed with LPSolve that mathematically computes the optimal solution. Even though LP solve performes well, it cannot be practically implemented because it uses maximization of equations that need a lot of resources and it would imply the use of expensive sensors.

Choosing one algorithm over another depends on the specific application. If the sensors have rechargeable batteries, the algorithm with prediction performs better due to the fact that it uses the concept of global maximums when choosing the best sensor to be turned on. The algorithm without prediction is better if the sensors have limited amount of energy because it makes use of local maxims, at the level of redundancy groups when choosing the sensor to recover the lost FoV. The advantages of each algorithm were showed and explained by test cases and results.

The simulations presented in this thesis were realized by a simulator, Simulo, which we implemented. The need of developing a new simulator came from the fact that the license free existing simulators do not offer the simulation possibilities needed in this thesis. Simulo is capable of performing both micro and macro simulation and also to implement several human behavior characteristics such as time response to an event, speedy drivers and also most of the driving aspects like overtaking on the left side for usual vehicles and on both sides for speedy drivers, coming back on the first lane if it is free, etc. The simulator is also capable of registering the simulation performed so that a comparison between algorithms with the exact test data sets to be possible. Furthermore Simulo is capable of simulating traffic flow specific for each lane, with different speeds, lengths, type and colors for each vehicle.

The overall idea of the thesis is that two performance efficient sensor dynamic management algorithms for data collection in the presence of obstruction in a WSN were proposed. They have solid mathematical background and were tested with a proposed simulator, Simulo, using Monte Carlo simulation. The algorithms proved their performance also by an important number of publications.

11.2 Report Contributions

This report has contributions on two directions: theoretical and practical contributions:

1. Theoretical:

   - The proposed metrics quantify in a good manner the researched area and represent the basis for a thorough analysis
   - An analysis of existing traffic monitoring algorithms
   - Mathematical background for the algorithm without prediction
   - Mathematical background for the algorithm with prediction
   - New algorithm for efficient traffic surveillance without prediction, using redundancy groups
   - New algorithm for efficient traffic surveillance with prediction, using event prediction

- Mathematical model for traffic simulation
- Algorithms performance from energy efficiency perspective (prolonging the lifetime of the network)
- Algorithms that offer coverage stability in the presence of dynamic obstacles
- The dynamical approach with the real time consideration of the different special situations that might appear (like the interruption of sensor communication or the energy depletion)
- By Monte Carlo Simulation, that represents the basis of the proposed traffic model, a certain empirical situation is generated according to some thorough developed mathematical models. This way a mathematical model, well researched both theoretical and practical that covers a large area of specific phenomena to the researched area is developed.

2. Practical:

- The implementation of the simulation framework Simulo
- The flexibility of the testing framework with respect to different traffic scenario and human behavior
- The processing of the data from the study in order to have real testing scenarios
- The automation of the simulation phase and post processing tasks
- The exemplification of real problems with suitable solutions

11.3 Future Work

Future work concerns both the development of the algorithms and also further development of Simulo.

One major work for the future is to implement the proposed algorithms into practice and compare the simulation results with the practical ones. Another aspect that shall be further developed is the dynamic adaptation of the network in case new sensors are added to the existing network. Furthermore, the performance of the algorithms in case the shape of the sensor changes, for example, circular sensors or radar sensors.

Regarding Simulo, further development implies introducing the possibility of testing traffic flow if on the lanes are different speed limits for certain areas or traffic lights. Also, at this moment Simulo is collision free, but the idea is to develop it to be able to also simulate collisions.

Also, at the network level, an important development would be have the possibility to dynamically add new sensors and these sensors to be immediately recognized by the network and also used by the algorithm that runs on the network.

Articles

All the work presented in chapters 5 and 6 was validated at:

1. . **I. Codruta**, D. Pescaru, A. Doboli, "Stochastic Model-based Heuristics for Fast Field of View Loss Recovery for Urban Traffic Management Through Networks of Video Cameras'', IEEE Transactions on Intelligent Traffic Systems, Accepted April, 2011, available online at: http://ieeexplore.ieee.org/search/srchabstract.jsp?tp=&arnumber=5741728&queryText%3Distin%26openedRefinements%3D*%26filter%3DAND%28NOT%284283010803%29%29%26searchField%3DSearch+All

2. **Codruta Istin**, Dan Pescaru, Horia Ciocarlie, Alex Doboli, "Monte-Carlo Simulation of a Dynamic Coverage Preservation Algorithm for Video Wireless Sensor Networks", in Proc. of the Fifth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX'10, Book Chapter in Monographs of System Dependability, Cap. 4, OVP Pub., ISBN 978-83-7493-526-5, Wrocław, Poland, June 29 – July 1, 2010, pag. 51-64. BOOK CHAPTER

3. **Codruta Istin**, Dan Pescaru Deployments Metrics for Video-based Wireless Sensor Networks, Scientific Bulletin of Politehnica University of Timisoara, Transactions on Automatic Control and Computer Science, ISSN 1224-600X, Vol. 52(66), No. 4, 2007, pp. 163-168. B+

4. Dan Pescaru, **Codruta Istin**, Daniel Curiac and Alex Doboli, Energy Saving Strategy for Video-based Sensor Networks under Field Coverage Preservation, in Proc. of the IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics (AQTR08), Cluj-Napoca, Romania, ISBN: 978-1-4244-2576-1, 22-25 May 2008, pp. 289-294. . ISI Proceedings

5. **Codruta Istin**, Dan Pescaru and Horia Ciocarlie, "Redundant Nodes Management in Wireless Sensor Networks", in Proc. of the 8th International Conference on Technical Informatics CONTI08, Vol. 2, ISSN 1844-593X, Timisoara, Romania, 5-6 June 2008, pp. 115-118.

6. D. Fuiorea, V. Gui, D. Pescaru, P. Paraschiv, **I. Codruta**, D. Curiac and C. Volosencu, Video-based Wireless Sensor Networks Localization Technique Based on Image Registration and SIFT Algorithm, WSEAS TRANSACTIONS on COMPUTERS, Issue 7, Volume 7, ISSN: 1109-2750, July 2008, pp. 990-999. ISI Proceedings

7. **Codruta** Istin, Dan Pescaru, Alex Doboli, Horia Ciocarlie, Impact of Coverage Preservation Techniques on Prolonging the Network Lifetime in Traffic Surveillance Applications, in Proc. of the 4th IEEE International Conference on Intelligent Computer Communication and Processing ICCP08, Cluj-Napoca, Romania, ISBN: 978-1-4244-2673-7, August 28 - 30, 2008, pp. 201-206. ISI Proceedings

8. **Codruta Istin**, Dan Pescaru, Alex Doboli, Horia Ciocarlie, Unidirectional and Omni-Directional Sensing Coverage Management in Wireless Sensor Networks, Scientific Bulletin of Politehnica University of Timisoara, Transactions on Automatic Control and Computer Science, ISSN 1224-600X, Vol. 53(67), No. 3, 2008, pp. 145-150. B+

9. **Codruta Istin**, Dan Pescaru, Horia Ciocarlie, Daniel Curiac, Alex

Doboli, Reliable Field of View Coverage in Video-Camera based Wireless Networks for Traffic Management Applications, IEEE Symposium on Signal Processing and Information Technology ISSPIT 2008, Sarajevo, Bosnia-Herzegovina, ISBN: 978-1-4244-3555-5, December 16-19, 2008, pp. 63-68. ISI Proceedings

10. M. Wang, V. Subramanian, A. Doboli, D. Curiac, D. Pescaru, **C. Istin**, "Towards a Model and Specification for Visual Programming of Massively Distributed Embedded Systems", IFSA Sensors and Transducers Journal, ISSN 1726-5479, Vol.5, March 2009, pp. 69-85.

11. Dan Pescaru, **Codruta Istin**, Florica Naghiu, Madalin Gavrilescu and Daniel Curiac, Scalable Metric for Coverage Evaluation in Video-based Wireless Sensor Networks, In Proc. of the 5th International Symposium on Applied Computational Intelligence and Informatics SACI09, Timisoara, Romania, ISBN 978-1-4244-4478-6, IEEE Cat. No. CFP0945C-CDR, May 28-29, 2009, pp. 323-328. ISI Proceedings

12. **Codruta Istin**, Dan Pescaru and Horia Ciocarlie,” Energy Saving Algorithm Tuning for Obstacle Avoidance Using Coverage Metrics for Video-Based Wireless Sensor Networks” ,4th IEEE International Conference on Intelligent Computer Communication and Processing ICCP09, Cluj-Napoca, Romania, ISI Proceedings

Bibliography

[1] Business Services Industry,DPAC Technologies Drives into Sensor Market with Airborne Wireless LAN Node Module, Sept 16, 2003

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", Journal of Computer Networks vol. 38, pp. 393–422, 2002

[3] Kazem Sohraby, Daniel Minoli, Taieb Znati, „Wireless Sensor Networks Technology, Protocols, and Applications", Published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2007

[4] Julien Basch , Leonidas J. Guibas , John Hershberger, "Data Structures for Mobile Data", Journal of Algorithms, 1998

[5] Raghavendra V. Kulkarni, Ganesh Kumar Venayagamoorthy, "Particle Swarm Optimization in Wireless Sensor Networks: A Brief Survey", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2011

[6] Kazem Sohraby, Daniel Minoli, Taieb Zanri, "Wireless Sensor Networks Technology, Protocols and Applications", Published by John Wiley &Sons, Inc., Hoboken, New Jersey, 2007

[7] Y. Yang, M. Cardei, "Movement-Assisted Sensor Redeployment Scheme for Network Lifetime Increase", in Proceedings of MSWiM'07, Chania, Greece, October 22–26, 2007.

[8] G. Wang, G. Cao, and T. F. LaPorta, "Movement-assisted Sensor Deployment", IEEE Transactions on Mobile Computing, vol. 5(6), pp. 640-652, 2006

[9] J. Wu and S. Yang, "SMART: a Scan-based Movement-assisted Sensor Deployment Method in Wireless Sensor Networks", in Proceedings of IEEE INFOCOM 2005, vol. 4, pp. 2313-2324, 2005.

[10] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility Limited Flip-based Sensor Networks Deployment", IEEE Transactions of Parallel and Distributed Systems, vol. 18(2), pp. 199-211, 2007.

[11] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self deployment algorithm for mobile sensor networks," Autonomous Robots, Special Issue on Intelligent Embedded Systems, Sep. 2002

[12] A. Savvides, C. C. Han and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," inProceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, 2001.

[13] T. He, C. Huang, B. Blum, J. A. Stankovic, T. Abdelzaher, "Rangefree localization schemes for large scale sensor networks," in Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom), San Diego, CA, 2003.

[14] E. Elnahrawy, X. Li and R. P. Martin, "The Limits of Localization Using Signal Strength: A Comparative Study," In Proceedings of The First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON 2004), Santa Clara, CA, USA, October 2004.

[15] Xin Liu, "Coverage with Connectivity in Wireless Sensor Networks", in Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems BROADNETS'2006, San Jose, USA, pp. 1-8, October 2006.

[16] U.M. Erdem and S. Sclaroff, "Optimal Placement of Cameras in Floorplans to Satisfy Task Requirements and Cost Constraints," in Proceedings of the fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, Omnivis2004, May 2004.

[17] C. Huang, Y.Tseng and L. Lo, "The Coverage Problem in Three Dimensional Wireless Sensor Networks," in Proceedings of the IEEE Conference Globecom, 2004, Vol.5, pp. 3182-3186.

[18] P. Bhattacharya, M. L. Gavrilova, "Voronoi diagram in optimal path planning", IEEE 4th International Symposium on Voronoi Diagrams in Science and Engineering, 2007.

[19] S. Megerian, F. Koushanfar, M. Potkonjak and M. B. Sirvastava, "Worst and Best-Case Coverage in Sensor Networks ", IEEE Transactions on Mobile Computing , vol 4, no 1, January 2005

[20] Yong Gao, Kui Wu, Fulu Li, "Analysis on the Redundancy of Wireless Sensor Networks", WSNA'03, San Diego, California, USA, September 2003.

[21] Cormen, T. and Leiserson, C. and Rivest, R. and Stein, C. Introduction to algorithms. In MIT Press and McGraw-Hill, 2001.

[22] Berman, P. and Calinescu, G. and Shah, C. and Zelikovsky, A. Power efficient monitoring management in sensor networks. In Proceedings of WCNC, 2004.
[23] Cardei, M. and Thai, M. and Li, Y. and Wu, W. Energy-efficient target coverage in wireless sensor networks. In Proceedings of INFOCOM, 2005.
[24] Brian Carter and Rammohan Ragade,"An Extensible Model for the Deployment of Non-Isotropic Sensors", IEEE Sensors Applications Symposium Atlanta, GA, February 12-14, 2008
[25] V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava, "Energy-Aware Wireless Microsensor Networks," IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 40-50, March 2002.
[26] Nilesh Khude, Anurag Kumar, and Aditya Karnik, "Time and Energy Complexity of Distributed Computation of a Class of Functions in Wireless Sensor Networks", IEEE Transactions On Mobile Computing, Vol. 7, No. 5, May 2008.
[27] Bender Paul, PhD Dissertation, "Energy Efficient Image Video Sensor Networks", Wright State University, 2008.
[28] Luiz H. A. Correia, Daniel F. Macedo, Daniel A. C. Silva, Aldri L. dos Santos, Antonio A. F. Loureiro, and Jos´e Marcos S. Nogueira. Transmission power control in mac protocols for wireless sensor networks. In MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pages 282. 289, New York, NY, USA, 2005. ACM.
[29] Habib M. Ammari and Sajal K. Das, "Trade-off between Energy Savings and Source-to-Sink Delay in Data Dissemination for Wireless Sensor Networks",MSWiM'05, October 10-13, 2005, Montreal, Quebec, Canada
[30] T. Shepard, "A channel Access Scheme for Large Dense Packet Radio Networks", in Proc of ACM SIGCOMM 1996, pp. 219-230, 1996
[31] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks," in Proceedings of the 6th Annual ACM/IEEE MobiCom, Boston, USA, August 2000
[32] S. Yoon and R. Dutta, and M. L. Sichitiu, "Power Aware Routing Algorithms for Wireless Sensor Networks," in Proceedings of the Third International Conference on Wireless and Mobile Communications ICWMC '07, 2007
[33] M. W. Subbarao, "Dynamic Power-conscious Routing for MANETS: an Initial Approach," in Proceedings of the IEEE Vehicular Technology Conference, pp. 1232–1237, 1999.
[34] I. Stojmenovic and X. Lin., "Power-aware Localized Routing in Wireless Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 11, pp. 1122–1133, 2001.
[35] V. Rodoplu and T. H. Meng, "Minimum Energy Mobile Wireless Networks," IEEE JSAC, vol. 17, no. 8, pp. 1333–44, Aug. 1999.
[36] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," HICSS, 2000.
[37] J. Aslam, Q. Li, and D. Rus, "Three Power-aware Routing Algorithms for Sensor Networks," Wireless Communications and Mobile Computing, vol. 2, no. 3, pp. 187–208, 2003.
[38] Q. Li, J. Aslam, and D. Rus, "Online Power-aware Routing in Wireless Ad-hoc Networks," in Proceedings of ACM/IEEE MobiCom2001, pp. 97–107 July 2001.
[39] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-efficient Coordination Algorithm for Topology Maintenance in Ad-hoc Wireless Networks," in Proceedings of ACM/IEEE MobiCom2001, July 2001.
[40] Jianping Pany Y. Thomas Houz Lin Caiy Yi Shiz Sherman X. Shen," Topology Control for Wireless Sensor Networks", MobiCom'03, September 14–19, 2003
[41] Nicholas H. Zamora, Jung-Chun Kao, Radu Marculescu, "Distributed Power-Management Techniques for Wireless Network Video Systems", EDAA 2007
[42] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-efficient Coordination Algorithm for Topology Maintenance in Ad-hoc Wireless Networks," in Proceedings ofACM/IEEE MobiCom2001, July 2001.
[43] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive Energy - Conserving Routing for Multihop Ad hoc Networks," Technical Report 527, USC/ISI, Oct.2000.

[44] X. Wang, L. Ding, D.W. Bi, and S. Wang, "Energy-efficient Optimization of Reorganization-Enabled Wireless Sensor Networks," Sensors Journal, vol. 7, pp. 1793-1816, 2007
[45] D. Tao, H. Ma1, and Y. Liu, "Energy-Efficient Cooperative Image Processing in Video Sensor Networks," PCM 2005, Part II, LNCS 3768, pp. 572 – 583, 2005.
[46] X. Wang, L. Ding, D.W. Bi, and S. Wang, "Energy-efficient Optimization of Reorganization-Enabled Wireless Sensor Networks," Sensors Journal, vol. 7, pp. 1793-1816, 2007
[47] D. Tao, H. Ma1, and Y. Liu, "Energy-Efficient Cooperative Image Processing in Video Sensor Networks," PCM 2005, Part II, LNCS 3768, pp. 572 – 583, 2005.
[48] M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior, "Cyclops: image sensing and interpretation in wireless networks," in Proceedings of the ACM SenSYS 2004, November 2004.
[49] D. Tao, H. Ma1, and Y. Liu, "Energy-Efficient Cooperative Image Processing in Video Sensor Networks," PCM 2005, Part II, LNCS 3768, pp. 572 – 583, 2005.
[50] K.-T. Song, J.-C. Tai, "Image-Based Traffic Monitoring With Shadow Suppression", Proceedings of the IEEE. Vol. 95 (2), 2007
[51] Y. Ando, et al., "Performance of Pheromone Model for Predicting Traffic Congestion", Proc. of AAMAS, 2006
[52] T. Rabie, B. Abdulhai, A. Shalaby, "Mobile Active-Vision Traffic Surveillance System for Urban Networks", Proc. Computer-Aided Civil and Infrastructure Engineering, 2005
[53] S. Soro and W. B. Heinzelman, "On the Coverage Problem in Videobased Wireless Sensor Networks", in Proceedings of the 2nd International Conference on Broadband Networks, vol. 2, pp. 932-939, 2005.
[54] Junaid Jameel Ahmad, Hassan Aqeel Khan, and Syed Ali Khayam," Energy Efficient Video Compression for Wireless Sensor Networks", CISS, 2009
[55] Paul Bender · Yong Pei," Development of Energy Efficient Image/Video Sensor Networks", Wireless Pers Commun, Springer, Nov 2008
[56] Rabie, T. and Abdulhai, B. and Ahalaby, A. Mobile active-vision traffic surveillance system for urban networks. In Proceedings of Computer-Aided Civil and Infrastructure Engineering, 2005.
[57] Braunstein, B. and Trimble, T. and Mishra, R. and Manoj, B. and Rao, R. On the traffic behavior of distributed wireless mesh networks. In Proceedings of International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), 2006.
[58] Szewczyk, R. and Osterweil, E. and Polastre, J., and Hamilton, E. and Mainwaring, A. and Estrin, D. Habitat monitoring with sensor networks. Communications of ACM, 47(6):34–40, 2004.
[59] Glaser, S. Some real-world applications of wireless sensor nodes. In Proceedings of SPIE Symposium on Smart Structures and Materials, 2004.
[60] Szewczyk, R. and Mainwaring, A. and Polastre, J. and Anderson, J. and Culler, D. An analysis of a large scale habitat monitoring application. In Proceedings of SenSys'04, 2004.
[61] He, T. and Krishnamurthy, S. and Stankovic, J. and Abdelzaher, T. and Luo, L. and Stoleru, R. and Yan, T. and Gu, L. Energy-efficient surveillance system using wireless sensor networks. In Proceedings of MobiSys, 2004
[62] Fehin, K. Adaptive traffic signals. are we missing the boat? ITE District 6 2004 Annual Meeting,
http://www.dksassociates.com/paper signals.asp, 2004.
[63] Papageorgiou, M. Traffic control. the handbook of transportation science. In R. W. Hall, Editor, Kluwer, 1999
[64] Ozlem Durmaz Incel and Amitabha Ghosh and Bhaskar Krishnamachari, "Scheduling Algorithms for Tree-Based Data Collection in Wireless Sensor Networks", Springer, 2011
[65] Nandini Prasad K S, G R Veerendra, Puttamadappa C, "Energy Efficient Data Gathering Scheme Using Cross Layer Cluster Approach For Wireless Sensor Networks", International Journal on Computer Science and Engineering, 2011
[66] Gartner, N. and Messer, C. and Rathi, A. Traffic flow theory: A state of the art report - revised monograph on traffic flow theory. U.S. Department of Transportation, Transportation Research Board, Washington D.C, 1992.

[67] Chrobok, R. and Wahle, J. and Schreckenberg, M. Traffic forecast using simulations of large scale networks. In Proceedings of IEEE Intelligent Transportation Systems Conference, 2001.
[68] Helbing, D. and Lammer, S. and Lebacque, J.-P. Self-organized control of irregular or perturbed network traffic. Optimal Control and Dynamic Games, 7:239–274, 2007.
[69] Richter, J. and Aberdeen, D. and Yu, J. Natural actor-critic for road traffic optimisation. In Proceedings of NIPS, 2006.
[70] Rigolli, M. and Brady, M. Towards a behavioral traffic monitoring system. In Proc. of AAMAS, 2005.
[71] Balmer, M. and Cetin, N. and Nagel, K. and Raney, B. Towards truly agent-based traffic and mobility simulations. In Proceedings of AAMAS, 2004.
[72] Dion, F. and Yagar, S. Real-time control of signalized networks - different approaches for different needs. pages, 56–60, 1996.
[73] Porche, I. A decentralized scheme for real-time optimization of traffic signals. pages, 582–589, 1996.
[74] Tomlinson, A. and Bull, L. Towards distributed adaptive control for road traffic junction signals using learning classifier systems. available at citeseer.ist.
[75] Henry, J. and Farges, J. and Tuffal, J. The prodyn real time traffic algorithm. In Proceedings of 4th IFAC-IFIP-IFORS Conference on Control in Transportation Systems, 1983.
[76] Gartner, N. A demand responsive strategy for traffic signal control. Transportation Research Record, (906), 1983.
[77] Yagar, S. and Han, B. A procedure for real-time signal control that considers transit interference and priority. Transportation Research, B28:315–331, 1984.
[78] Felici, G. and Rinaldi, G. and Sforza, A. and Truemper, K. Intelligent traffic control: A logic programming approach and a real application. Transportation Research Part C: Emerging Technologies, 14(3):175–189, June 2006.
[79] Sims, A. and Dobinson, K. Scat: the sydney coordinated adpative traffic system philosophy and benefits. pages, 19–41, 1971
[80] Tavladakis, K. and Voulgaris, N. Development of an autonomous adaptive traffic control system. In Proceedings of European Symposium on Intelligent Techniques, 1999.
[81] Mauro, V. Road network control. pages, 361–366, 1991.
[82] Fuck, O. and Zemk, P. and Tupec, P. and Crha, L. and Herout, A. The networked photo-enforcement and traffic monitoring system unicam. In Proceedings of IEEE International Conference and Workshop on Engineering of Computer-Based Systems, 2004.
[83] Ando, Y. and Fukazawa, Y. and Masutani, O. and Iwasaki, H. and Honiden, S. Performance of pheromone model for predicting traffic congestion. In Proceedings of AAMAS, 2006.
[84] Cardei, M. and Du, D. Improving wireless sensor network lifetime through power aware organization. page, 2005.
[85] Cardei, M. and MacCallum, D. and Cheng, M. and Min, M. and Jia, X. and Li, D. and Du, D. Wireless sensor networks with energy efficient organization. page, 2002
[86] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, " Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks ", in ACM SIGMOBILE Conference on Mobile Computing and Networking, Rome, Italy, July 2001.
[87] www.autodesk.com/autocad
[88] Belomestny, D., Jentsch, V., Schreckenberg, M. "Completion and continuation of nonlinear traffic time series: a probabilistic approach". Journal of Physics A: Math. Gen. 36, 2003 11369–11383
[89] Lauritzen, S.L. "Graphical Models", Oxford University Press, 1996
[90] Tomas Singliar and Milos Hauskrecht, "Modeling Highway Traffic Volumes", Proceedings of the 18th European conference on Machine Learning, 2007
[91] Hastie, T., Tibshirani, R., Friedman, J. " Elements of Statistical Learning", Springer, 2001
[92] Jensen, F.V., "An Introduction to Bayesian Networks", Springer-Verlag, New York, 1996

[93] Istin C., Pescaru D., Ciocarlie H., Curiac D., Doboli A., Reliable Field of View Coverage in Video-Camera based Wireless Networks for Traffic Management Applications, In: IEEE Symposium on Signal Processing and Information Technology ISSPIT, 2008.
[94] Istin C., Pescaru D., Ciocarlie H., Doboli A., "Monte-Carlo Simulation of a Dynamic Coverage Preservation Algorithm for Video Sensor Networks", In: Fifth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX 2010
[95] Nagel, Kai; Schreckenberg, Michael, "A cellular automaton model for freeway traffic", Journal de Physique I, Volume 2, Issue 12, December 1992, pp.2221-2229
[96] Baykal-Gursoy, Duan Z., Xu H., "Stochastic Models of Traffic Flow Interrupted by Incidents", Control in Transportation Systems, 2010
[97] Baykal-Gürsoy M. and Duan Z., "Stochastic decomposition of queues in a random environment", ISE Working Paper, 07-022, 2007
[98] Helbing D., "A section-based queuing-theoretical traffic model for congestion and travel time analysis in Networks", J.Phys.A. 36, L593, 2003
[99] Richards P. I., Shock waves on the highway, " Operations Research", 4, 42-51, 1956
[100] Helbing.d, Hennecke.A, and Treiber.M, " Phase diagram of traffic states in the presence of inhomogeneities", Phys. Rev. Lett. 82, 4360, 1999
[101] Prigogine I. and Andrews F.C., "Boltzman-like approach for traffic flow, Operations Research" 8(6), 789-797, 1960
[102] Helbing D., "A section-based queuing-theoretical traffic model for congestion and travel time analysis in Networks", J.Phys.A. 36, L593, 2003.
[103] Lammer S., Donner R., and Helbing D., "Anticipative control of switched queueing systems", Euro. Physics.J.B 63, 341-37, 2008
[104] Lighthill M. J. and Whitham, G. B., "On kinematic waves: II. A theory of traffic on long crowded roads", Proc. Roy. Soc. London Ser. A 229, 317-345, 1955.
[105] Kuhne.R.D., in Proceedings of the 10th International Symposium on Transportation and Traffic Theory, p.119, 1987
[106] Dominic Jost, Kai Nagel, "Probabilistic Traffic Flow Breakdown In Stochastic Car Following Models", Traffic and Granular Flow '2003
[107] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics" PhD thesis, University of Cologne, Germany, 1997. See www.zpr.uni-koeln.de
[108] Papageorgiou M. Traffic control. the handbook of transportation science. In: R. W. Hall, Editor, Kluwer, 1999.
[109] John Stanley, " Passenger Car Equivalents Of Trucks Under Lane Restriction And Differential Speed Limit Policies On Four Lane Freeways", PhD thesis, 2009
[110] Paul Croney, David Croney, "The design and performance of road pavements", third edition, MkGraw Hill, 1997.
[111] http://www.racbusiness.co.uk/useful-contacts/fleet-statistics/
[112] Martin Treiber and Arne Kesting, "An Open-Source Microscopic Traffic Simulator", IEEE INTELLIGENT TRANSPORTATION SYSTEMS MAGAZINE VOL. 2, NO. 3, 6-13, 2010
[113] Jérôme Härri, Marco Fiore , Fethi Filali Christian Bonnet "Vehicular Mobility Simulation with VanetMobiSim", SIMULATION, Vol. 00, Issue 0, Xxxxxxxx 2009 000–000 c_ 2009 The Society for Modeling and Simulation International DOI: 10.1177/0037549709345997
[114] Tomer Toledo, "Integrated Driving Behavior Modeling", PhD thesis, 2003
[115] Vanetmobisim project home page, http://vanet.eurecom.fr
[116] Vanet simulator sumo, http://sumo.sourceforge.net/
[117] Francisco J. Martinez, Chai Keong Toh, Juan-Carlos Cano, Carlos T. Calafate and Pietro Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)", WIRELESS COMMUNICATIONS AND MOBILE COMPUTING Wirel. Commun. Mob. Comput. (2009) Published online in Wiley InterScience