Market-based Framework for Mobile Surveillance Systems

by

Ahmed Elmogy

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Ahmed Elmogy

Abstract

The active surveillance of public and private sites is increasingly becoming a very important and critical issue. It is therefore, imperative to develop mobile surveillance systems to protect these sites. Modern surveillance systems encompass spatially distributed mobile and static sensors in order to provide effective monitoring of persistent and transient objects and events in a given Area Of Interest (AOI). Mobile sensors have emerged as a solution to overcome the limitations of the static sensors. Mobile sensors are capable of sensing, processing, moving and communicating with other nodes. They can sample the environment at different locations, exchange information with other nodes, and collaboratively accomplish the required mission. Mobile surveillance systems incorporate self-organized networks of mobile sensing nodes of different modalities, data and information fusion nodes, acting nodes and control nodes. These self-organized nodes can collaboratively and continuously sense within the volume of interest, as well as physically manipulate and interact with it. These surveillance systems provide systematic observation of an AOI that includes the timely detection, localization, recognition and identification of objects and events, their relationships, activities, and plans, in order to determine whether they are behaving normally, or whether there is any deviation from their expected behavior. To achieve this complete situation awareness, the system starts by collecting the relevant data in order to identify situation entities and their relationships. Then the system performs a relational analysis of object-events, followed by intent estimation and consequence prediction.

The realization of the potential of mobile surveillance requires the solution of different challenging problems such as task allocation, mobile sensor deployment, multisensor management, cooperative object detection and tracking, decentralized data fusion, and interoperability and accessibility of system nodes.

This thesis proposes a market-based framework that can be used to handle different problems of mobile surveillance systems. Task allocation and cooperative target-tracking are studied using the proposed framework as two challenging problems of mobile surveillance systems. These challenges are addressed individually and collectively.

Although a great number of task allocation approaches have been reported in the literature, many aspects have to date been given only sparse attention. Examples of these aspects are complex task allocation and constrained task allocation. The developed surveillance framework in this thesis addresses complex task allocation that uses task tree structure. Hierarchical and centralized complex task allocation algorithms are presented to allocate a set of complex tasks to a mobile sensor team.

Besides complex task allocation, the cooperative target tracking problem is also tackled in this thesis. The proposed cooperative tracking approach is inspired by market-based economic systems. In the proposed framework, mobile sensors are self-interested agents with the primary goal of maximizing individuals' profits. The presented methodology inherits the flexibility of markets in allowing cooperation and competition to emerge opportunistically among the agents. These agents implement an Extended Kohonen Maps-based algorithm for target tracking. Two versions of the Extended Kohonen Maps are proposed: supervised, and unsupervised. A hybrid clustering technique is used to decrease the number of active trackers during tracking, and thus save energy.

Acknowledgements

I'd like to express my sincere thanks and appreciation to my supervisor, Professor Fakhri Karray for all his guidance, support, and patience throughout the course of this work. I am grateful for the opportunity to work with and to learn from such an extraordinary teacher.

I am also indebted to Professor Alaa Khamis for offering me valuable advice, support, and encouragement during the course of this work. I am extremely fortunate to have benefited from his experience.

I wish also to thank Professors William Melek, Pin Han Ho, and Shawki Ariebi for serving as my advisory committee. I am thankful for all their advice and comments. Many thanks to professor Abdulmotaleb El Saddik for serving as my external examiner.

I also want to express my thanks to the Robotics and Autonomous Systems (RAS) focus group members: Bahador Khaleghi, Miao Yun Qian, Allaa Hilal, Mehrdad J. Gangeh, Manglai Zhou, Karim El-Rayes, and Rodrigo Araujo for their constructive feedback during our regular group meetings.

I would also like to thank the Egyptian government for its great financial support during four years of my PhD work.

Dedication

To my wife who has given me her warmest support, and to my parents who have given me the greatest love and care.

Contents

List of Tables **xii**

List of Figures **xv**

List of Symbols **xvi**

1 Introduction **1**

1.1 Motivations and Objectives 1

1.2 Contributions 4

1.3 Thesis Statement 6

1.4 Thesis Organization 6

2 Mobile Surveillance Systems **8**

2.1 Introduction 8

2.2 Multisensor Surveillance 11

2.3 Mobile Surveillance Systems 13

2.4 Challenging Problems of Mobile Surveillance Systems 15

2.4.1 Situation Awareness 15

2.4.2 Sensor Management 16

2.4.3 Task Allocation 18

2.4.4 Cooperative Target Detection and Tracking 20

2.5 Organizational Paradigms of Mobile Surveillance Systems 21

2.5.1 Centralized Architecture 22

2.5.2 Decentralized Architecture 23

2.5.3 Hierarchical Architecture 24

2.5.4 Holonic Architecture 26

2.6 Problem Solving Strategies 27

2.6.1 Information-theoretic Approach 27

2.6.2 Control-theoretic Approach 29

2.6.3 Optimization Approaches 30

2.6.4 Market-based Approach 31

2.7 Conclusion 32

3 Market-based Approach **34**

3.1 Introduction 34

3.2 Auctions 35

3.3 Types of Auctions 36

3.3.1 Single-good Auctions 36

3.3.2 Combinatorial Auctions 39

3.3.2.1 Winner Determination Strategy 40

3.4 The Pros and Cons of Market-based Approaches 45

3.5 Concluding Remarks 47

4 Task Allocation in Mobile Surveillance Systems 48

4.1 Related Work 49

4.2 Complex Task Allocation 52

4.2.1 Problem Definition 52

4.2.2 Problem Formulation 53

4.3 Proposed Market-based Approach 55

4.3.1 Single-shot and Combinatorial Auctioning 55

4.3.2 Auction Design 57

4.3.2.1 Utility Function 57

4.3.2.2 Search Tree 57

4.3.3 Allocation Levels 61

4.3.3.1 Point-Level Allocation 61

4.3.3.2 Area-Level Allocation 61

4.3.3.3 Mission-Level Allocation 62

4.3.4 Winner Determination Strategies 63

4.3.4.1 Centralized Allocation 64

4.3.4.2 Hierarchical Allocation 64

4.4 Fixed and Dynamic Tree Allocation 65

4.4.1 Fixed Tree Task Allocation 66

4.4.2 Dynamic Tree Task Allocation 75

4.5 Conclusion 81

5 Target Detection and Tracking 82

5.1 Introduction 82

5.2 A Classification Taxonomy of Current Target Tracking Approaches 83

5.2.1 The Number of Trackers versus the Number of Targets 84

5.2.2 Tracker Mobility 85

5.2.3 Environment Complexity 86

5.2.4 Type of Cooperation 86

5.2.5 Coordination of Multiple Trackers 87

5.2.6 Evaluation Criteria 87

5.3 Related Work 88

5.4 General Problem Statement 93

5.5 Problem Formulation 94

5.6 Approach 96

5.6.1 Target Detection 96

5.6.2 Find the Center of Gravity 97

5.6.2.1 *Subtractive Clustering* 98

5.6.2.2 *K-means Clustering* 99

5.6.3 Extended Kohonen Neural Network 101

5.6.3.1 Supervised Extended Kohonen Neural Network 104

5.6.3.2 Unsupervised Extended Kohonen Neural Network 106

5.6.4 The Proposed Cooperative Multi-target Tracking Approach 107

5.7 Conclusion 111

6 Experimental Results 113

6.1 Complex Task Allocation Simulations and Results 113

6.1.1 Outdoor Scenario 114

6.1.2 Indoor Scenario 118

6.2 Target Tracking Simulation and Results 122

6.2.1 Tracking using Supervised Learning EKM 124

6.2.2 Tracking using Unsupervised Learning EKM 125

6.2.3 Cooperative Multi-target Tracking Simulations 128

6.3 Conclusion 134

7 Conclusion 136

7.1 Conclusion 136

7.2 Future Work 138

Bibliography 158

List of Tables

2.1 Organizational architectures comparison [1] [2] 28

6.1 Test Cases 115

6.2 Average cost for outdoor task allocation 116

6.3 Average cost for indoor task allocation 121

6.4 Energy saving comparison 132

List of Figures

1.1 Market-based framework for mobile surveillance systems 5

2.1 Multisensor surveillance system [3] 12

2.2 Endsley SA model 17

2.3 Centralized architecture 22

2.4 Decentralized architecture 24

2.5 Hierarchical architecture 25

2.6 Holonic systems 26

3.1 Auction taxonomy 37

3.2 Search on items 41

3.3 Search on bids 42

3.4 Depth first search 43

3.5 Breadth first search 44

4.1 Single-shot and combinatorial auctioning 56

4.2 Surveillance scenario 59

4.3 *AND/OR* task tree 60

4.4 Decompose-then-allocate approach 62

4.5 Allocate-then-decompose approach 63

4.6 Higher-level allocate-then-decompose approach 64

4.7 Centralized auctioning 65

4.8 Hierarchical auctioning 66

5.1 Field of view (a)Omnidirectional (b) Conic 97

5.2 Kohonen Map 102

5.3 Extended Kohonen Map 104

5.4 supervised Extended Kohonen Map learning 105

5.5 Cueing/Handoff market-based algorithm 109

6.1 Comparison of the average cost for centralized allocation mechanism for outdoor environment 117

6.2 Comparison of the average cost for hierarchical allocation mechanism for outdoor environment 118

6.3 A comparison of the solution quality of task tree allocation varying the number of mobile sensors 119

6.4 A comparison of the solution quality of task tree allocation varying the number of areas 120

6.5 Comparison of the average cost for centralized allocation mechanism for indoor environment 122

6.6 Comparison of the average cost for hierarchical allocation mechanism for outdoor environment 123

6.7 single target tracking using supervised learning EKM 126

6.8 single target tracking using unsupervised learning EKM 127

6.9 comparison between EKM average coverage for different n and m ratios . . 129

6.10 comparison between EKM average energy saving for different n and m ratios 130

6.11 comparison between EKM and other approaches 131

6.12 comparison between EKM and CMOMMT average coverage for $n = m$ . . 132

6.13 Waterloo airport Surveillance 133

6.14 Mission fulfillment time for Waterloo airport surveillance 134

List of Symbols

h	:	the number of inputs of a linear system defined by state space model
q	:	the number of outputs of a linear system defined by state space model
$n1$	:	the number of state variables of a linear system defined by state space model
$X(t)$	:	the state variables of the state space model
$U(t)$	:	the input variables of the state space model
$Y(t)$	:	the output variables of the state space model
A_s	:	the state, the input, output, and feedforward matrices of the state space model
B_s	:	the input matrix of the state space model
C_s	:	the output matrix of the state space model
D_s	:	feedforward matrix of the state space model
M	:	a set of items to sell
B	:	a set of bundle bids
$Se_j \in M$	:	is a set of items from the big set M
p	:	the price at which an item is sold
S	:	a set of mobile sensors
$T = \{t_1, ...t_n\}$	:	a set of tasks
A	:	a function that maps T to S
G	:	a group of tasks that is decomposable into other tasks
Md	:	the tasks that result from decomposing G

Bf : a function that maps the decomposed tasks Md to S

U : a set of sensors utilities

u_{ij} : is the utility of sensor i to execute task j

$b_s(t)$: the bid of mobile sensor s on task t

$b_s(G)$: the bid of mobile sensor s on bundle of tasks G

β : a function that describes the characteristics of the bundle G

$C(A)$: the total cost required to execute the allocation A

G_s : the bundle of tasks that is won by sensor s

$p_s(t)$: the total payment a mobile sensor s receives after executing the task t

$d_s(t)$: the total distance a mobile sensor s travels to reach the task

$Auctid$: the auctioneer ID

$cost(T)$: the cost of executing a task tree T

T_{auctid} : the task tree owend by the auctioneer

O : a set of n moving targets

$o_j(tt)$: the position of of target o_j at time tt

$SC(s_i, tt)$: the sensor coverage, which is observable by mobile sensor s_i at time tt

$OM(tt)$: $m \times n$ Observation matrix

AC : the average covrage of the tracking algorithm

$g(OM(tt), j)$: a function that decides if a target j is covered or not

$(xx_1, ..xx_{n3})$: data points to be clustered

D : the density required for subtractive clustering

xx_{c_1} : the initial cluster centre of the subtractive clustering

D_{c_1} : the largest density value of the subtractive clustering

r_a, r_b : positive numbers used in subtractive clustering

J : the K-means clustering objective function

c_i : the K-means clusters centres

MM : the K-means membership function

GR_i	:	the K-means group number
w_r	:	the fan-in weight vector associated to the neuron placed at position r on Extended neural grid
w_{ne}	:	fan-in vector of the neuron ne
z_{ne}	:	fan-out vector of the neuron ne
X	:	input data space
Y	:	output data space
$< x; y >$	:	a set of examples applied to the Extended neural network
z_{ne}^{old}	:	the fan-out weights of the winner ne before learning
z_{ne}^{new}	:	the fan-out weights of the winner ne after learning
α	:	neural network learning rate
(θ_{ne}, dis_{ne})	:	the input weights of the neuron ne
$speed(ne)$	:	the speed value of the winning neuron ne
$angle(ne)$	:	the angle value of the winning neuron ne
$v = (\theta_s, dis_s)$	:	the actual displacement of the tracker in target tracking algorithm
$Guas(ne, i)$	:	the Gaussian function of the distance between the winning neuron and input neurons
VOI	:	volume of interest
SM	:	sensor management
SA	:	situation awareness
MAS	:	multi-agent system
DFS	:	depth first search
BFS	:	breadth first search
A*	:	best first search
IDA*	:	iterative deeeping best first search
AOI	:	area of interest
RFID	:	radio frequency identification

MRTA	:	multi-robot task allocation
OAP	:	optimal assignment problem
ST	:	single-task robots
MT	:	multi-task robots
SR	:	single-robot task
MR	:	multi-robot task
IA	:	instantaneous assignment
TA	:	time-extended assignment
ST-SR-IA	:	single task robot-single robot task-instantaneous assignment
TSP	:	traveling salesman problem
SSP	:	shortest sequence planning algorithm
STST	:	single tracker single target
STMT	:	single tracker multiple targets
MTST	:	multiple trackers single target
MTMT	:	multiple trackes multiple trackers
JPDA	:	joint probabilistic data association
DJPDA	:	decoupling joint probabilistic data association
PDA	:	probabilistic data association
VTT	:	a virtual target tracking
CMOMMT	:	cooperative multi-robot observation of multiple moving targets

Chapter 1

Introduction

1.1 Motivations and Objectives

Many autonomous tasks require sensory data to be collected from sensors that are able to sample the environment at different locations, exchange the information with other nodes, and collaboratively accomplish the required mission. This type of sensors is called mobile sensors. The coordination and control of multiple mobile sensors provides an opportunity to improve the quality and robustness of the collected data, as compared to a single sensor and/or static system [4]. In other words, well-designed interactions between the mobile sensors can ensure good performances and produce collective intelligence at the group level. Multisensor systems have been the focus of multiple research efforts in the last few decades. This can be seen in many applications such as space and underwater exploration [5, 6], search and rescue [7, 8], surveillance and reconnaissance [9, 10].

In the last few years, major terrorist attacks such those of September 11, 2001 have led to an active research focused on surveillance. One of the hot topics is how to automate surveillance tasks based on mobile and fixed sensors platforms [11]. Many benefits can be anticipated from the use of multisensor systems in surveillance applications, such as decreasing task completion time, and increasing mission reliability. Advanced surveillance systems include a vast array of cooperative (static and mobile) sensors with varying sensing modalities that can sense continuously the volume of interest [12]. These sensors are dynamically used based on the current and future states of the environment. The main goal of the surveillance system is to adjust the sensing conditions for improved visibility, and thereby improve performance [13]. In such setting, surveillance is a complex problem posing many challenging problems.

Generally, monitoring of public and private sites is the main application of multisensor surveillance systems. Surveillance operation includes the timely detection, localization, recognition and identification of objects and events, their relationships, activities, and plans, in a given Volume of Interest (VOI) [14]. There are many factors that increase the complexity of surveillance operations such as high tempo, high density in certain environments- like school campuses and shopping malls- and collateral damage in application with dense activity such as in military operations [15]. The primary objectives of the surveillance systems are to provide the information that makes the system able to understand and predict the actions and the interactions of the observed objects in order to carry out different tasks. Examples of these tasks would include target search, identification, and tracking.

This thesis presents a market-based framework for mobile surveillance systems. The goal is to develop a framework that efficiently distributes tasks among the mobile sensor team to achieve the surveillance mission. Such a framework will support the operation of the mobile sensors so that they can collaboratively perform tasks such as detecting and tracking moving targets. In order to maximize the effectiveness of the mobile sensor team collaborating as a group, the action of every mobile sensor should consider the contribution of its teammates towards the mission objectives. How to accomplish this is a complex problem, which is currently an active area of research [5, 16–19].

This thesis addresses the problem defined by the following question:

How can a system composed of multiple mobile sensors be distributed in a given environment and how can functionality be configured, utilized and coordinated in order to efficiently accomplish the surveillance mission.

This problem requires tackling some of the challenging problems of mobile surveillance that are still open. These problems include, but are not limited to, task allocation, mobile sensor deployment, multisensor management, cooperative object detection and tracking, decentralized data fusion, and interoperability and accessibility of system nodes. This thesis tackles two of these problems: task allocation, and cooperative detection and tracking.

The surveillance problem addressed in this thesis can be cast as one of finding a mechanism that accepts representations of robot states and abilities and outputs a desirable team behavior.

The problem of task allocation in mobile surveillance systems addresses the question of finding the task-to-sensor assignments that optimize global cost or utility objectives [20, 21]. This can be divided into two sub-problems. First, how a set of tasks is assigned to a set of mobile sensors. Second, how the behavior of the sensor team is coordinated in order to achieve the cooperative tasks efficiently and reliably. A great number of task allocation approaches have been reported in the literature. This is mainly seen in surveillance, reconnaissance, and search and rescue applications [7–10]. However, many aspects have to date been given little attention. Examples of these aspects are complex task allocation, and constrained task allocation. Complex tasks are those tasks which can be decomposed into different subtasks. Working with a complex task is guaranteed to produce a more efficient solution for the task allocation problem [19]. Thus, complex task allocation is considered in this thesis as a part of the design of the mobile surveillance framework.

The recent interest in surveillance in public, military, and commercial scenarios is increasing the need to autonomously observe the movements of targets navigating in a bounded area of interest. The target-tracking problem is concerned with estimating the state of a target or targets in some areas of interest based on some measurements from the stationary sensors, in most of the cases. However, using stationary sensors has many constraints which force the use of multiple mobile sensors dynamically moving over time especially in surveillance applications. Examples for these constraints are the lack of prior information about the area of interest, and large areas to be observed that economics prohibit the placement of a large number of static sensors. Thus, target detection and tracking is also tackled in this thesis using mobile sensors in the design of the mobile surveillance framework.

Our approach to solving the surveillance problem is inspired by market-based economic systems. Market mechanisms create an environment within which individual self-interested agents are given incentives to lead them to achieve a good system-wide solution [22]. These market mechanisms are continually gaining considerable attention and popularity due to several desirable properties [18]. The most desirable property of the market mechanisms is efficiency. Using market mechanisms gives the opportunity to focus on performance goals while letting the market algorithms innovate to achieve these goals at lowest cost [22].

1.2 Contributions

This section highlights the major contributions to this research activity to research areas of task allocation, target tracking, and mobile surveillance systems as illustrated in Fig. 1.1. Detailed description of the proposed framework components are provided in the next chapters as shown in Fig. 1.1. The major contributions in this thesis are summarized as follows:

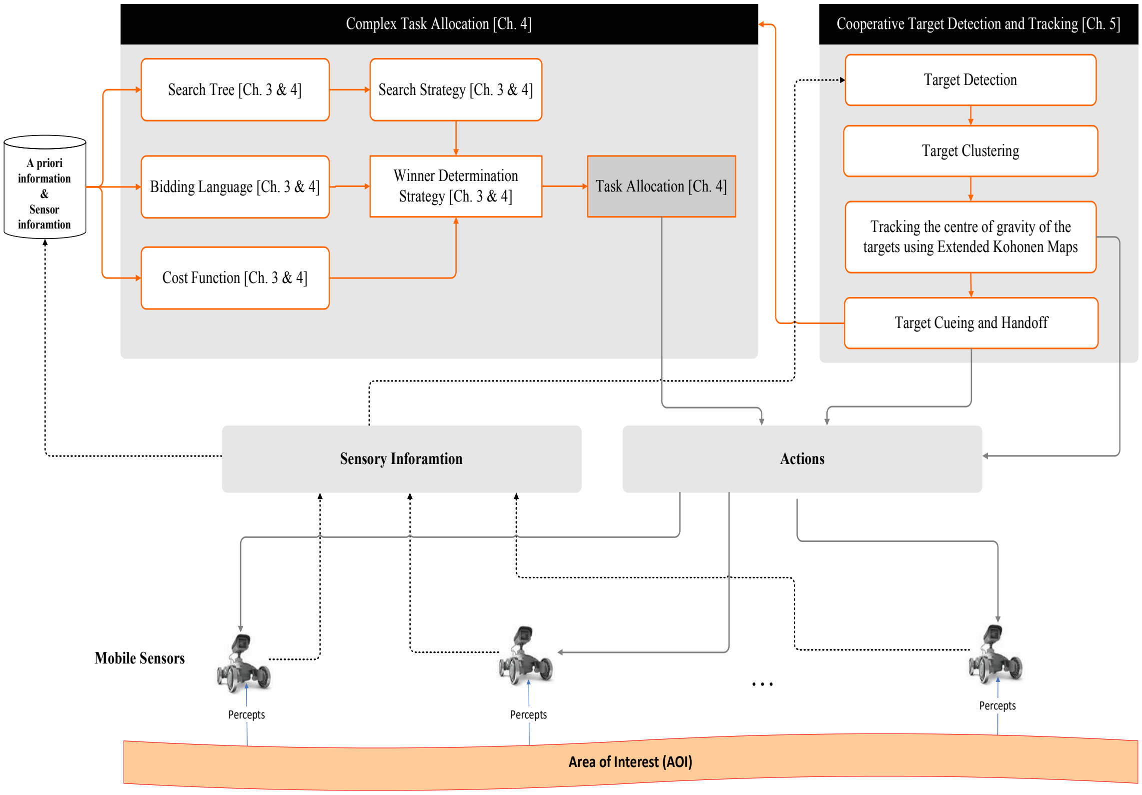

Complex Task Allocation [Ch. 4]
Cooperative Target Detection and Tracking [Ch. 5]
Search Tree [Ch. 3 & 4]
Search Strategy [Ch. 3 & 4]
Target Detection
A priori information & Sensor inforamtion
Bidding Language [Ch. 3 & 4]
Winner Determination Strategy [Ch. 3 & 4]
Task Allocation [Ch. 4]
Target Clustering
Tracking the centre of gravity of the targets using Extended Kohonen Maps
Cost Function [Ch. 3 & 4]
Target Cueing and Handoff
Sensory Inforamtion
Actions
Mobile Sensors
Percepts
Percepts
…
Percepts
Area of Interest (AOI)

Figure 1.1: Market-based framework for mobile surveillance systems

1. A market-based mobile surveillance framework is presented which will tackle various issues that have not been investigated collectively in the past. These include; task allocation problem, target search and detection, and target tracking. The benefits and challenges of each issue are assessed. The proposed framework is applied to indoor and outdoor environments

2. A complex task allocation approach is proposed for incorporation into the design the mobile surveillance framework. This approach uses a tree structure as a representation of tasks to be allocated. This provides the surveillance framework the ability to allocate the tasks to the mobile sensor team at different levels of abstraction. Two types of task tree are developed; a pure *AND* tree, and *AND/OR* tree. Centralized

and hierarchical dynamic and fixed tree task allocation algorithms are developed to solve the multisensor task allocation problem.

3. An Extended Kohonen neural network for the purpose of target tracking is developed. Two types of networks are proposed: supervised learning, and unsupervised learning networks. An energy-efficient methodology to coordinate the behaviour of the sensor team during tracking is presented. The core concept of this methodology is the introduction of target clustering before tracking so as to reduce the number of mobile sensors needed to track the moving targets. A hybrid subtractive − K-means clustering technique is used for this purpose. Also, a market-based algorithm is developed as a coordination methodology between sensor team during tracking.

1.3 Thesis Statement

This thesis asserts that the use of market-based techniques in the mobile surveillance systems enables efficient and robust task allocation and cooperative target tracking.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 gives a global overview over mobile surveillance systems. For that, the challenging problems, the organizational paradigms, and the problem solving strategies that are observed in the design of multi-sensor surveillance are introduced in brief.

Chapter 3 discusses in details the market-based approach describing different types of auctions focusing on combinatorial auctions. The knowledge given in this chapter helps to understand the different aspects of market-based approaches.

Chapter 4 presents a market-based approach to solve the multi-sensor task allocation problem in mobile surveillance systems. Centralized and hierarchical algorithms with fixed and dynamic trees have been examined, focusing on complex tasks.

Chapter 5 presents a detailed description of the developed methodology for tracking multiple objects to be incorporated into the proposed mobile surveillance framework. The Extended Kohonen Map is proposed for single target tracking and multi-target tracking.

Chapter 6 gives the experimental setups, experimental results, and discussion.

Chapter 7 summarizes this work with conclusions and future directions. It also lists the publications originated from the research work in this thesis.

Chapter 2

Mobile Surveillance Systems

The main purpose of this chapter is to give a global overview of mobile surveillance systems. It is divided into six major sections. Section 2.1 gives an introduction to multi-sensor systems, and their applications. A brief discussion of the mobile surveillance systems and their operations is given in sections 2.2 & 2.3. A brief discussion of the challenging problems of mobile surveillance, as seen in the literature, is presented in section 2.4. This is followed by the organizational paradigms that are observed in the design of mobile surveillance systems in 2.5. The problem solving strategies are presented in section 2.6. Finally, important conclusions are drawn in section 2.7.

2.1 Introduction

Monitoring of public places is increasingly becoming very critical, especially after the World Trade Center, and Pentagon attacks in September 2001 [23]. This increased the need to create and deploy intelligent and automated surveillance systems [24]. This can be accomplished using either standard surveillance systems or advanced surveillance systems, depending on the complexity of the monitoring process. Standard surveillance systems are good in situations where it is only required to detect and track people moving in the observed scene. Advanced surveillance systems fit best in applications that require understanding of complex human behavior, or detection and recognition of faces. These advanced systems are based on multiple sensors (cameras, laser, infrared, thermal, radar, etc.).

Surveillance operations include the timely detection, localization, recognition and identification of objects and events, their relationships, activities, and plans, in a given Volume of Interest (VOI) [14]. In application domains such as security or defense, this process is referred to as picture compilation. The aim of this is to generate a representation of the area under surveillance (e.g., a port) based on data and information from a variety of sources. The compiled picture essentially displays information about red objects (adversary, non-cooperative), blue objects (own and friendly), white objects (neutral, usually cooperative), and environmental conditions. The following are a few factors that further define the complexity of surveillance operations [25]:

- **High tempo**: The limited time available to understand the impact of the events on the mission at hand, and to react to them. High tempo imposes the requirement that critical (potentially red) objects be detected as early as possible so as to provide more reaction time to the human decision makers.

- **High density**: Certain environments, such as school campuses, shopping centers, airports, etc., exhibit significant congestion with dense activity (e.g., commercial, educational and recreational traffic), as compared to open uncrowed environments. High density necessitates increased effort on the part of the system to gain and maintain situation awareness, thus distracting it from focusing on critical objects.

- **Collateral damage**: In critical applications, collateral damage also imposes a non-uniform environment that cannot be pictured as a confrontation between blue (friendly) objects and red (enemy or undesirable) objects. Blue and red, as well as neutral (white), objects are interspersed and overlapping, presenting a highly complex challenge with respect to discerning one type of object from another. This situation increases the risk of undesirable effects, e.g., casualties.

Automated surveillance systems provide a promising solution for these challenges.

The need of automated surveillance systems in public places, airports, commercial, law enforcement, and military applications has been greatly increased in the last few decades. Conducting surveillance in these applications requires using the team concept with a considerable number of resources. Automated monitoring in the surveillance systems addresses the real time observation of people and vehicles within a busy environment, thus leading to a complete description of their actions and interactions [26]. These actions and interactions can be classified as follows [27]:

- Object class: such as people, and vehicles
- Object action: such as appearing, moving, stopped, and disappearing
- Object interaction: such as close, moving away, moving toward

Surveillance systems have been seen in many applications. These applications include but not are restricted to:

1. **Public Security**: The main goal of these applications is to ensure security for citizens. This will be done by providing efficient, user-friendly, highly automated surveillance of crowded areas such as airports, shopping malls, and railways [15, 28].
2. **Public Health Surveillance**: Public health surveillance provides information needed to guide interventions. For example, if the objective is to prevent the spread of epidemics of acute infectious diseases, managers need to intervene quickly to stop the

spread of this disease. Therefore, they need a surveillance system that provides rapid early warning information from clinics and laboratories.

3. **Health care**: Visual surveillance can be employed to help handicapped persons and to monitor the activities of elderly people [23].

4. **Military Applications**: As the governments are trying to increase the agility and versatility of their weapon systems, they need to adopt intelligent surveillance systems that are able to get information in order to support the weapon systems and the operating methods against enemies and/or other threats [29]. Such systems are intended to detect, locate and track moving targets, which include humans, trucks, tanks, etc.

It is unrealistic to monitor several areas by one sensor, or to track a moving object for a long period of time. So using multisensor in the surveillance system is a must.

2.2 Multisensor Surveillance

Multisensor systems have become an active research area and are highly prominent in several new application areas in the recent years. These applications include but are not limited to space and underwater exploration [5, 6], search and rescue [7, 8], surveillance and reconnaissance [9, 10], etc. Many benefits can be anticipated from the use of multisensor systems such as resolving complexity by decomposing the complex task into simple tasks, decreasing task completion time, and increasing mission reliability. Thus having only one mobile sensor may work as a bottleneck for the whole system, especially in critical times.

In surveillance systems, there is an increasing need for more accurate and reliable sensors as they are continuously applied to new, more complex and demanding tasks. A sensor network is one of the frameworks that have been proposed recently [30] to improve the capabilities of a sensing system. Such a network, consists of multiple similar and/or dissimilar sensors that work in tandem to perform a common surveillance mission. There are three general types of sensors that can be used in multisensor systems [13]:

1. **Complementary sensors**: which are able to perceive different features using different sensors devices. These features can be combined to produce more detailed information than would be available from any single sensor.

2. **Cooperative sensors**: which are able to provide information that can't be derived from only one sensor.

3. **Competitive sensors**: which perceive the same measurements from the environment in order to reduce uncertainty of the observed data and thus, increase the accuracy of the surveillance system.

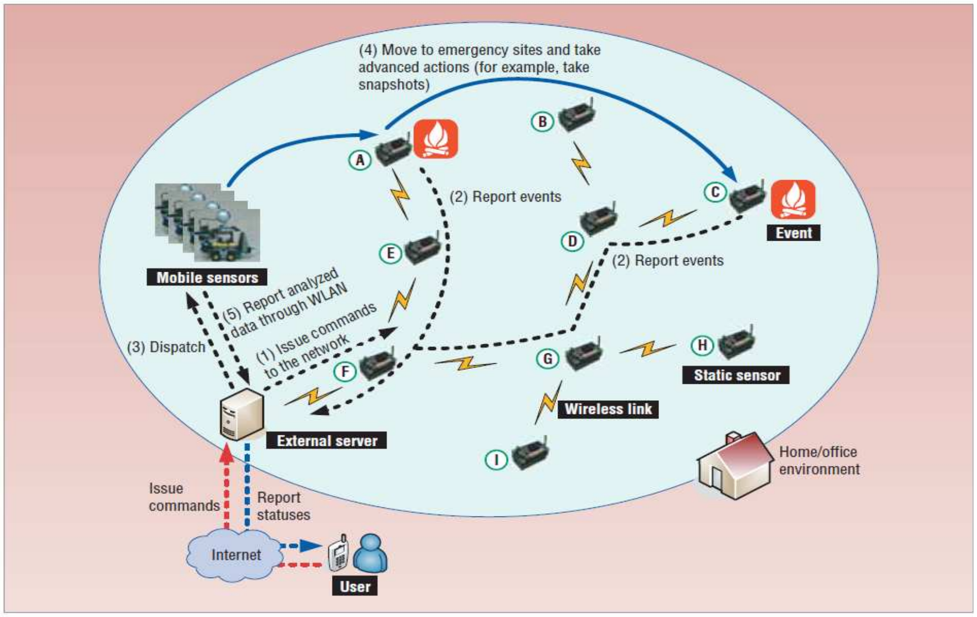


Figure 2.1: Multisensor surveillance system [3]

Monitoring of public and private sites is increasingly becoming a very important issue in the past few years [23]. In order to achieve that, a multi-sensor surveillance system should be developed. Multi-sensor surveillance systems consist of different types of sensors such as cameras, laser finders, and radars, which can be installed on fixed or mobile devices [31]. Using multi-sensors gives the surveillance system the ability to augment the capabilities of single sensor in many aspects [23]. The primary aims of these surveillance systems are to provide the required information that makes the system able to understand and predict the actions and the interactions of the observed objects in order to carry out different tasks (search, scanning, identification, and tracking) [32, 33]. An example of such systems is shown in Fig. 2.1 [3]. Each mobile sensor is equipped with basic motor, sensory and communication modules that make its design simple. Well-designed interactions between these simple sensors can ensure complex performances and produce collective intelligence at the group level. One of the potential applications of multisensor system is surveillance. Advanced surveillance systems include a vast array of cooperative (static and mobile) sensors with varying sensing modalities that can continuously sense the volume of interest [34].

.

2.3 Mobile Surveillance Systems

Effective monitoring of persistent and transient objects and events is a key to the effective protection of any Area Of Interest (AOI). Surveillance is the systematic observation of AOI by visual, audio or other means. This systematic observation includes the timely detection, localization, recognition and identification of objects and events, their relationships, activities, and plans, in a given AOI in order to determine whether they are behaving normally, or if there is any deviation from their expected behavior. To achieve this complete situation awareness, the system starts by collecting the relevant data in order to identify situation entities and their relationships. Then the system performs a relational analysis of object-events followed by intent estimation and consequence prediction [35] .

Mobile surveillance systems incorporate self-organized networks of mobile sensing nodes, data and information fusion nodes, acting nodes and control nodes. These self-organized nodes can sense collaboratively and continuously AOIs and physically manipulate and interact with them. The sensing nodes represent a set of spatially distributed mobile sensors of different modalities that can sense collaboratively and continuously an AOI. These nodes include but are not limited to vision systems, sonar/infrared/laser range finders, microphone arrays or RFIDs mounted on mobile bases in order to overcome the limitations of the static sensors. Mobile sensing nodes are capable of sensing, processing, moving and communicating with other nodes. They can sample the environment at different locations, exchange the information with other nodes, and collaboratively achieve the required mission. Fusion nodes combine information from the sensing nodes about the targets and events in order to determine whether they are behaving normally or if there is any deviation from their expected behavior. Actuation may be a direct physical action upon the process, such as moving a camera to keep track of an agile target; or a physical making of an electrical circuit, which in turn has a direct effect upon the process. An example would be an actuator (relay) that activates an alarm, a fire extinguisher or hard-kill/soft-kill weapons. Control nodes manage sensing, fusion and acting nodes to provide timely detection, localization, recognition and identification of targets and events, their relationships, activities, and plans, in a given AOI.

2.4 Challenging Problems of Mobile Surveillance Systems

In this section, the challenges that face the design of mobile surveillance systems will be discussed. Four important challenges will be studied: situation awareness, sensor management, task allocation, and cooperative target detection and tracking.

2.4.1 Situation Awareness

SA(situation awareness) is defined as the system's ability to remain aware of everything that is happening and at the same time to integrate that sense of awareness into what it is running at the moment [36]. Having complete, accurate and up-to-the date SA is essential where technological and situational complexity overwhelming the human decision-maker is a concern [37, 38]. When people are required to make critical choices sometimes at a fast pace the vast majority of errors that occur are a direct result of failures in situation awareness. In the operation of complex systems, the result can be a catastrophic airplane crash, emergency response teams flounder, and critical commercial systems falter all at a great cost in lives and money [39].

The three-level model of situational awareness was developed by Endsley [38] to understand aviation tasks, such as piloting aircraft and air traffic control. In these types of tasks, people are required to keep up-to-date with a dynamically changing environment. Endsleys model is arranged into three hierarchical levels of situational assessment [38], each stage being a necessary precursor to the next level. This model illustrates three steps for situation awareness formation as shown in Fig. 2.2:

- **Perception**: The first step in achieving SA. This level is used to perceive the status, attributes, and dynamics of relevant elements in the environment. In other words, this stage involves the processes of monitoring, detection, and recognition. This will lead to an awareness of multiple situational elements (objects, events, people, environmental factors), and their current states (locations, conditions, modes, actions).

- **Comprehension**: This level may follow on from the perception of the elements and their current states. The perceived data can be integrated to produce an understanding of these elements. Comprehension is essential to understand the significance of the elements, and to gain a picture of what is going on in the surveillance system.

- **Projection**: The highest level of SA, which involves the ability to project the future actions of the elements in the environment. This can be achieved through knowledge of the status and dynamics of the elements (perception), and comprehension of the situation, and then extrapolating this information forward in time to determine the future states of the surveillance system.

2.4.2 Sensor Management

In mobile surveillance systems, distributed sensing resources can provide cooperatively complete, accurate, and timely information about the presence and activity of all objects or events within the area of interest when properly managed. Sensor management (SM) is defined as the process that plans and controls the use of a set of sensors in a manner that synergistically improves the process of data fusion and ultimately that of perception and understanding. In [40], SM is the process that deals with various problems, including allocation, coordination, and scheduling. A more generic statement of the SM will be to allocate, coordinate, and schedule the sensor usage to accomplish a specific and often dynamic mission. Allocation is concerned with the selection of the most suitable sensor(s) for each task. Coordination focuses on the efficient use of sensors by managing the interaction among the sensors in order to minimize conflicts and maximize synergy. In other words,

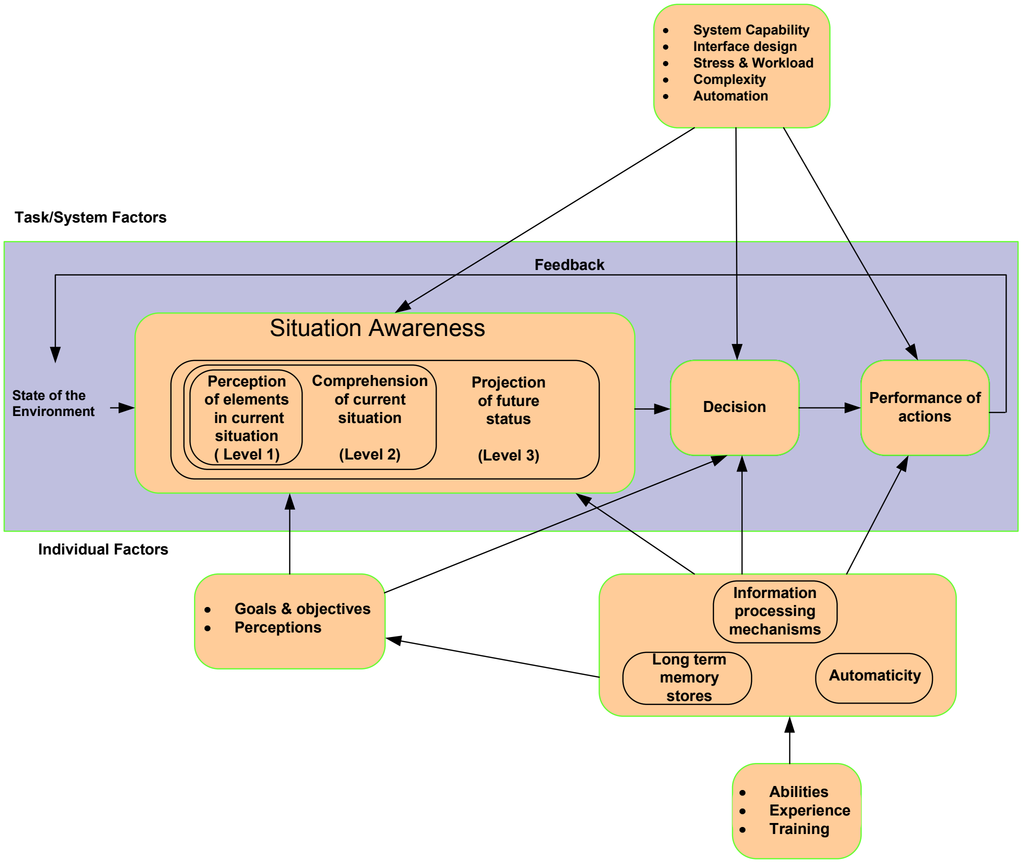


Figure 2.2: Endsley SA model

SM must determine which sensor is most important for the mission and thereby prevents the others sensors from operating, or otherwise set up a schedule to allow one sensor to operate for a period of time, and then some other sensors for another period. The coordination may also consist in having different sensors work together to acquire information on a common object/event [41]. This, for instance, consists of dynamically tasking some sensors to fill the coverage gaps of other sensors. Finally, scheduling is the designation of time segments for specific tasks or activities, the nature of which is defined during the allocation or coordination tasks.

The ultimate goal of the SM is to optimize the overall performance of the multi-sensor system [42]. This goal is based on predefined system goals and priorities. These goals and priorities are defined by the user requirement. Examples for that are: to maintain good track quality at a predefined sector, and to identify or provide more observations on an unknown high threat target. SM seeks to answer the following questions [43]: which sensor or sensors, which task, how to control the sensors, and when to start. The role and function of SM can be best understood if structured into different levels based on their functionality [42]:

- **Level 1**: This level involves individual control of each sensor, such as direction, pointing, change in frequency, power level, etc.
- **Level 2**. This level focuses more on the sensor tasks and different modes of the sensor with respect to the operational needs. This stage needs algorithms to prioritize the tasks, and to determine when and how a mode should be activated. Thus sensor task scheduling, sensors cueing (sensor handoff, target acquisition by another sensor aiding), and sensor modes are implemented at this level.
- **Level 3**. This is the highest level in SM, where dynamic sensor placement, and effective/optimal sensor mix should be addressed.

2.4.3 Task Allocation

One of the fundamental issues that arises in mobile surveillance systems is how to assign a set of tasks to a set of mobile sensors to effectively perform a given system level task. This tasks-sensors assignment is called task allocation process. This process may need to be continuously adjusted to adapt for the changes in the environment and/or group performance. This makes dynamic task allocation one of the essential challenges for mobile surveillance systems. For a group of mobile sensors to effectively perform a system level task, the mobile sensors will coordinate to distribute the tasks amongst themselves in a way that enables them to accomplish their mission efficiently and reliably [16, 44]. Thus task allocation is a twofold problem. First it addresses how to assign a set of tasks to a set of mobile sensors. Second it considers how to coordinate the behavior of the sensors team in order to do the cooperative tasks efficiently. The past decade has witnessed a growing emphasis in research topics highlighting task allocation [20, 45–48]. Established techniques can be leveraged to help tackle task allocation aspects. Although numerous techniques exist for these aspects [20, 45, 46], they exhibit high degrees of complexity which makes the task allocation problem a significantly *NP* class problem.

The Task allocation problem is an exciting problem and it can be considered in many areas of the daily life. This thesis will focus on the mobile surveillance application as one of the rich domains that presents task allocation challenges. The research efforts in this area have revealed that accomplishing the subtasks in order to do the surveillance level task is not easy. Adding to this difficulty, the mobile sensors that are assigned these subtasks need to be coordinated to achieve the surveillance mission efficiently. An example of a surveillance problem is as follows:

A team of mobile sensors has been dispatched to aid in the task of scanning different areas of interest looking for dangerous targets. These areas of interest can be viewed from multiple vantage points for a more accurate assessment of the situation. A coalition of mobile sensors may be formed in order to share scanning of some areas. Also, some areas are completely scanned by only one mobile sensor. When the sensor discovers that an area may contain some targets, a task is defined which requires tracking these targets. A sensor or a coalition of sensors may be needed to perform the tracking mission. Sometimes some sensors need assistance for what they are doing: for example, a team member can help to track some targets, which cannot be continuously tracked by other sensors. The difficulty of these tasks provides some of the basis why this research is important. This thesis discusses in details this problem and proposes a market-based solution as shown in section 4.3.

2.4.4 Cooperative Target Detection and Tracking

The recent interest in surveillance in public, military, and commercial scenarios is increasing the need to autonomously observe the movements of targets navigating in a bounded area of interest. Thus assigning one or more mobile sensors to follow and approach a target from multiple directions and angles to obtain images will improve the knowledge about the target. This may allow improved target detection, and identification. Generally, the combined coverage capabilities of the available sensors will be insufficient to cover the entire terrain of interest, which force the use of multiple sensors dynamically moving over time.

The problem of target tracking is to estimate the state of a target based on some measurements from sensors. The estimation process may contain some uncertainties for several reasons such as noise measurement, unauthenticated measurement origin, and unknown target movement. The performance of a tracking system can be improved by exploiting multiple sensors [49], however, this requires the coordination of movement strategies for cooperative target tracking. The uncertainty of the estimation of target position can be reduced by combining measurements from multiple sensors [50]. In this case, the movement strategy of each sensor is to minimize the redundant measurement uncertainty. An example of this is when the horizontal and vertical uncertainty of a single sensor measurement is different; two sensors can minimize the total estimation. The target tracking problem can be studied from different points of view [51], such as the number of trackers, the number of targets, the mobility of trackers, and the type of cooperation between trackers.

The target tracking problem can be mainly divided into two categories: non-ccoperative target tracking, and cooperative target tracking. In the non-cooperative target tracking problem, there is no cooperation among the trackers, even when there are multiple trackers involved in a system, whereas in the cooperative target tracking system, single or multiple targets are tracked by multiple trackers cooperatively. In this case, the effect of cooperation can be maximized by designing a suitable coordination strategy. This coordination strategy modifies the behavior of the tracker either directly or by other trackers. Although the target tracking problem has been investigated thoroughly, most research in this area has focused on the development of a non-cooperative tracking algorithms. However, in security or surveillance applications, it is often more important to carefully coordinate multiple sensors in order to cover a large structured environment. Motivated by this, this thesis will focus on the development of an energy-efficient target tracking algorithm which is able to accomplish the tracking mission cooperatively by coordinating the mobile sensors to be incorporated into the mobile surveillance framework.

2.5 Organizational Paradigms of Mobile Surveillance Systems

As mentioned previously, mobile surveillance systems encompass a set of spatially distributed mobile sensing nodes or agents. These agents must be governed by a well-defined organizational paradigm that determines the roles, relationships, and structures [52]. It has been repeatedly shown that the organization of a system can have significant impact on its short and long-term performance [53]. Generally speaking, there is no single paradigm that is suitable for all applications. Some paradigms are more suitable for some applications and less suitable for other applications. In this section, different organizational paradigms of the multi-agent systems are introduced. The merits and demerits of each paradigm will be presented in brief.

2.5.1 Centralized Architecture

The basic structure of centralized system is shown in Fig. 2.3. In this type of systems, each agent maintains a connection to one central agent. Thus, the separate agents send all the information they have to the central agent, which in turn processes this information and sends the appropriate commands to these agents to control their operations. The advantages of this type include the reduction of duplication of effort, resources, and increaed savings of cost and time [54]. Although the centralized systems are widely implemented in the literature [55], there are many disadvantages that restrict the use of this paradigm in many applications. The lack of robustness is one of the most important disadvantages of the centralized system. In other words, if the central agent fails, the whole system will fail. Also, the system scalability is restricted because all the agents are connected to the central agent.

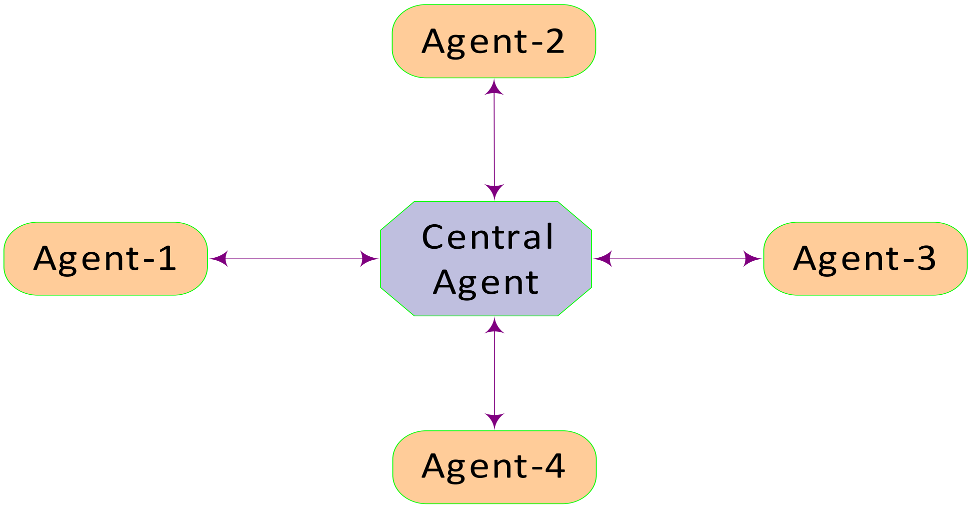


Figure 2.3: Centralized architecture

In [56], a centralized surveillance system is presented. This surveillance system is mainly developed to track targets in some areas of interest. For that, two stages are proposed. The first stage is the optimization of a network of agents containing ultrasonic sensors within a pre-specified coverage area. In the second stage, the mobile sensors are activated based on need according to the tracking requirement of the moving object. The mobile sensors have the capability to communicate with a centralized server to report on target location and direction. Another example for centralized systems can be found in [57]. The authors in this work proposed a surveillance system that utilizes data fusion from multiple types of sensors. This surveillance system is designed for monitoring an indoor environment.

2.5.2 Decentralized Architecture

Decentralization is the process of dispersing the administrative tasks and authorities between the agents of the multi-agent system [54]. A possible configuration of the decentralized systems is shown in Fig. 2.4. In this type of configuration, all the agents in the system are connected to each other. Thus, each agent is communicating its information with the other agents. Each agent can work on its own without major consideration of the other agents. Also, sometimes an agent of the decentralized system needs to exchange information with other agents in order to achieve its mission efficiently in tandem with other agents. The main advantage of the distributed system is its robustness. In other words, if one of the agents fails, the other agents are still working on their own and/or cooperatively with others. The robustness of the distributed systems gives this type of systems the capacity to add other agents. This means that scalability is no longer an issue in decentralized systems.

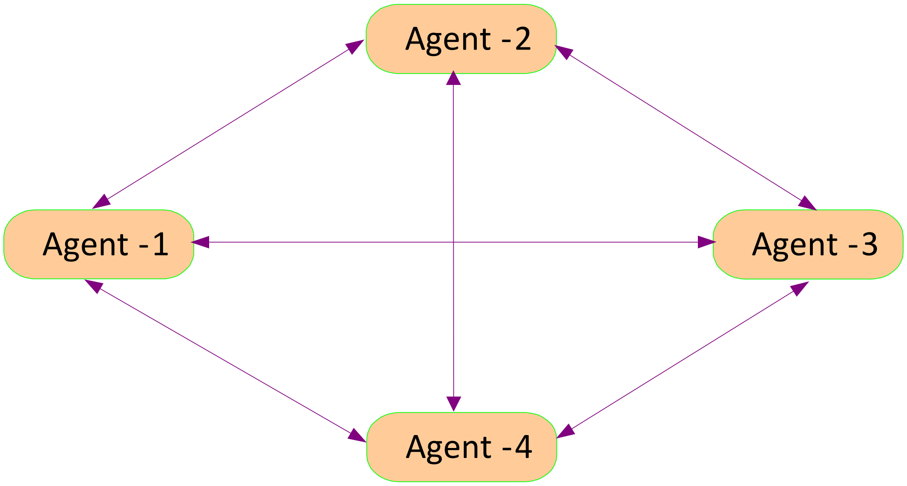


Figure 2.4: Decentralized architecture

An example of a decentralized surveillance systems can be found in [58]. In this work, a multisensor surveillance system is proposed. The proposed system achieves a high degree of survivability by employing a decentralized sensing architecture. The main goal of this work is to track people and robots as they move around a factory room. Also, in [59], an agent-based negotiable game theoretic approach for decentralized sensor management is presented. Sensor assignment occurs locally, and thus there is no central node of failure. This approach provides a means of developing a fully decentralized architecture for tracking and surveillance systems that eliminates computational bottlenecks

2.5.3 Hierarchical Architecture

In this type of systems, the agents are arranged in a hierarchical tree structure. An example of a hierarchical systems is shown in Fig. 2.5. A hierarchy can link agents either directly or indirectly [52]. Commands, tasks and goals to be achieved flow down from central agents to subordinate agents, whereas sensations and command results flow up from subordinate to central agents. Most of the multi-sensor applications require the decomposition of the tasks into many tasks, which makes a hierarchical structure a good choice for these applications.

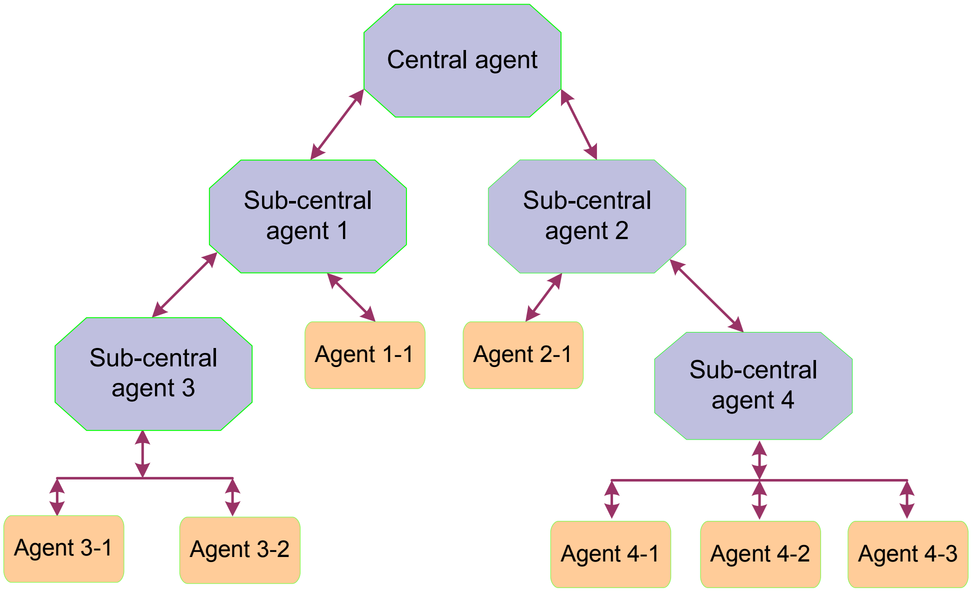


Figure 2.5: Hierarchical architecture

Generally, hierarchical architecture has been used to solve surveillance system problems such as target tracking, and sensor management. In [60], a scalable hierarchical multiple target tracking algorithm is proposed. The task of tracking is done hierarchically by forming a tracking group around a supernode, and later combining tracks from different supernodes. Also, in [61], a hierarchical sensor management scheme is proposed to safeguard the data packet passing on the sensor network under different types of attacks. The proposed scheme also contains group communication policies, group membership requirements and an algorithm for generating a distributed group key for secure communication.

2.5.4 Holonic Architecture

From the multi-agent systems (MASs) point of view, holonic systems are composed of self-reliant agents that are capable of achieving a specified flexible behavior. More specifically, a holon can be thought of as a special type of agent that is characteristically autonomous, cooperative and recursive [62]. An example of such a system is shown in Fig. 2.6 [52]. Each group (circle) in this system has a character derived but distinct from the members of the group. At the same time, this group contributes to the properties of one or more groups above it. The structure of these groups is a basic unit of organization that can be seen throughout the system as a whole. In this context, the holonic systems are suitable to simulate and model complex systems without multiple granularities [63].

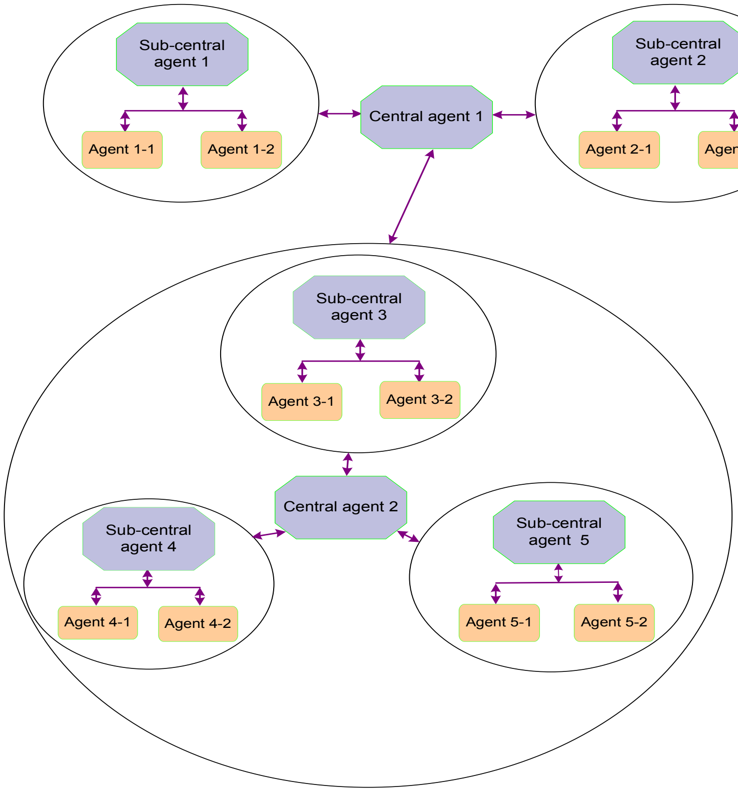


Figure 2.6: Holonic systems

In [64], a conceptual design for sensor management in military settings using holonic control is presented. The main role of holonic control in this work is to improve tactile sensor management for military surveillance operations. Also, a holonic multi agent approach is presented in [65] for solving the coverage problem while monitoring some areas of interest. In this holonic approach, a group of sensors can behave like one atomic entity, therefore the patrol of sensors can be built using the holonic concept.

The pros and cons of the aforementioned organizational paradigms can be summarized in table 2.1:

2.6 Problem Solving Strategies

As explained previously, designing a framework for mobile surveillance systems is very difficult because of the different challenges and the difficulties the design should pass with. In this section, some of the most important problem solving strategies that can be used to design a framework for mobile surveillance systems will be briefly discussed.

2.6.1 Information-theoretic Approach

The main concept of information theory is that of entropy [66]. The information-theoretic entropy concept has arisen from the long-standing lack of a solid basis for the understanding of randomness [67]. Entropy is a measure of the uncertainty associated with a random variable. In this context, the term by itself usually refers to the Shannon entropy [66], which quantifies the information contained in a message, usually in units such as bits in the sense of an expected value. Equivalently, the Shannon entropy is a measure of the average information content one is missing when the value of the random variable is not known.

Table 2.1: Organizational architectures comparison [1] [2]

Architecture	Advantages	Disadvantages
Centralized	Can achieve optimal performance(theoretically) Simplicity	Communication bottleneck potential Reliability issue Lack of modularity and scalability
Distributed	Promises higher scalability Robustness, and efficiency	Difficult to design and control
Hierarchical	Reduces communication and/or computational overhead taking a top-down decomposition of tasks	Expensive to adapt in dramatically dynamic situations
Holonic	Enables the construction of very complex sytems Adaptable to changes in environments	Difficult to design and control

Generally speaking, an information-theoretic approach allows the formulation of a rigorous mathematical model to describe large scale systems. This occurs particularly in situations where uncertainty is prevalent, and where the system at hand is decentralized [68]. Many tools are provided by an information theoretic approach, such as Bayesian analysis, data fusion and decentralized decision making. Several information-theoretic approaches have been implemented to consider surveillance systems problems. An example of this can be found in [69], in which an information-theoretic approach is utilized to measure detection or estimation performance. The primary problems of interest in this proposed approach are multiple object tracking and identification, sensor network management, and multifunction radar control.

2.6.2 Control-theoretic Approach

The control theory is the branch of engineering and mathematics that deals with dynamic systems. The most famous representation of dynamic systems using control theory is state space. State space is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. The state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time. The state space representation provides a convenient and compact way to model and analyze systems with multiple inputs and outputs. To abstract from the number of inputs, outputs and states, the variables are expressed as vectors and the differential and algebraic equations are written in matrix form. Laplace transforms are used to encode all the information about a control system. The use of the state space representation is not limited to systems with linear components and zero initial conditions. The most general state-space representation of a linear system with h inputs, q outputs and $n1$ state variables is written in the following form:

$$\begin{aligned} \dot{X(t)} &= A_s X(t) + B_s U(t) \\ Y(t) &= C_s X(t) + D_s U(t) \end{aligned} \tag{2.1}$$

where $X(t)$, $U(t)$, and $Y(t)$ are the state, input, and output variables matrices whereas A_s, B_s, C_s, and D_s are the state, input, output, and feedforward matrices.

Some of the surveillance systems problems have been solved using control-theoretic approaches. In [70], a decision theoretic approach is proposed for dynamic sensor selection. The proposed approach is used to allocate the available surveillance cameras to some urban areas. The main goal is to monitor these areas to localize and track people in these areas. Also, in [71], a hybrid control approach is proposed for cooperative target tracking. This control approach is used to steer mobile vehicles to a region near the target with a certain orientation condition, and then coordinate their motion to cooperatively capture the target through an enclosing circular motion around the target.

2.6.3 Optimization Approaches

Optimization theory is the study of the minima and maxima values of an objective function. In other words, determining the most profitable or least disadvantageous choice out of a set of alternatives for that function [72]. The set of alternatives is restricted by several constraints on the values of a number of variables and the objective function locates the optimum in the remaining set. The objective function is the mathematical expression that combines some variables to express a system goal. Topics in optimization theory range from conditions for the existence of unique minima and maxima values to methods for finding these values.

Many optimization approaches have been used to solve surveillance systems problems. In [73], an approach that is based upon genetic algorithms is used for accurately locating and identifying groups of targets engaged in a common activity. The long-term goal of this approach is to develop an understanding of how to define a group and then to understand the inter-relationships between the various characteristics that describe a group. This has been used as an objective to design a military surveillance system. Also, in [74], an algorithm based on ant colony optimization is used for solving sensor management problem. The interest in solving the sensor management problem comes from its widespread use in multisensor surveillance systems.

2.6.4 Market-based Approach

Market-based approaches are those which inherit the behavior of auctions. Auctions are the economic activities that exist in many places in everyday situations, in several forms. An auction provides a mechanism to allocate a set of goods to a set of bidders based on their bids. Auctions may be categorized into two main categories; single item auctions [75], and combinatorial auctions [19]. In single item auctioning (sometimes called sequential auctioning), the items are auctioned one at a time and the auctioneer always allocates that item to the highest bidder. But if the bidders are interested in a combination of items, then it is very difficult for the bidders to submit bids, because they do not know what items they will receive in a later auction. In combinatorial auctions, multiple goods are auctioned simultaneously. This type of auctioning helps to overcome the inefficiencies in allocation due to related uncertainties [76]. In other words, the value of an item that a bidder wins depends greatly on the winning of other items.

Many market-based algorithms have been proposed for several application domains. These problems include but are not restricted to sensor management [77], task allocation [16], and job scheduling [78]. These market-based approaches are continuously gaining considerable attention and popularity because of several desirable properties [22]. The most desirable property of the market based approaches is efficiency. Auctions are able to produce an efficient solution for many objective functions. Furthermore, market-based approaches are robust to several types of failures because of their usual independency of a central agent. The efficiency and robustness of the market-based approaches make them good prospects to be scalable. Thus, market-based approaches are still able to produce efficient solutions, even when the agents' size and/or inputs increase. However, market approaches do have some disadvantages [18]. Negotiation protocols, which are the main theme of the market-based approaches can complicate the design of the market architecture, and increase communication requirements. Chapter 3 discusses this approach in more detail.

2.7 Conclusion

In this chapter, a review of mobile surveillance systems has been given. Five important sections have been studied. In section 2.1, an introduction to multisensor systems and their applications has been presented. A brief discussion of mobile surveillance systems and their operations has been given in sections 2.2 & 2.3. The challenging problems of multisensor surveillance as seen in the literature have been introduced in section 2.4. This has been followed by a brief discussion of the organizational paradigms of multisensor surveillance systems in section 2.5. Problem solving strategies have been presented in section 2.6. Some conclusions can be drawn from this chapter:

1. The first is that the design of mobile surveillance systems is challenging. These types of systems can be studied from different points of view, such as challenging problems, organizational paradigms, and problem solving strategies.

2. It also seems that most of the current body of research in the field of mobile surveillance systems focuses on tackling these systems across various issues with several

techniques, but without investigating how to integrate these issues into a unified surveillance system framework.

Chapter 3

Market-based Approach

Market-based approaches have received significant attention and are growing very fast in the last few decades, especially in multiagent domains [17, 19, 21, 79, 80]. Motivated by this great attention, a market-based framework for mobile surveillance systems is presented in this thesis. As explained in the remaining chapters of this thesis, the proposed framework deals with two main problems of mobile surveillance systems: namely, complex task allocation and cooperative target detection and tracking. Both problems have been handled using market-based approaches. This chapter serves as a vehicle to understand different aspects of market-based approaches. The chapter discusses in details the market-based approach, describing different types of auctions focusing on combinatorial auctions. It also highlights the pros and cons of the market-based approach.

3.1 Introduction

Market-based task allocation is an economically-inspired approach that provides a way to coordinate the activities of a number of competitive agents. The approach imitates the auction process of buying and selling services through bidding [81]. Sellers or auctioneers are responsible for processing the bids sent by buyers or bidders, and for determining the winning bidder.

In a market-based approach, an economy is nothing more than cooperation between populations of agents to produce a global output [82]. Thus, agents coordinate with each other to produce an aggregate set of goods or tasks. The market-based mechanisms are descriptive, which is different from the theoretic mechanisms, which are normative. These mechanisms use the agents' outcomes to determine which strategies will be followed to do the required tasks efficiently. In other words, the agents make deals among themselves in order to achieve the requested mission more efficiently than they could do with original distribution of the resources and tasks.

A market-based approach is suitable when a system consists of multiple self-interested mobile agents. Each agent possesses private information that is relevant to solving a system-wide problem, without the need for a global view. In this case the allocation process is similar to a sealed first-price auction where bidders submit one bid in a concealed fashion. A buyer's sealed offer is a function of a buyer's reservation value, a price beyond which the buyer would be better off without the transaction. The bidder with the highest price wins and pays that price [75]. The following section provides more details about the auctioning process, which is the core of market-based approach.

3.2 Auctions

An auction is a process of buying and selling goods or services by offering them up for bid, taking bids, and then selling the item to the highest bidder. Auctions are widely used in real life, in consumer, corporate, and government settings. This makes auctions very important to consider when tackling many applications. More fundamentally, auctions provide a general theoretical framework for understanding resource allocation among self-interested agents [83]. Generally, any protocol that allows agents to indicate their interest in one or more resources or tasks is considered an auction. Thus, auctions can be considered in many applications that might not even use money as the basis of payments as normally thought. Also, auctions provide a way to compress the information in the form of bids [84], which increases the efficiency in terms of both the amount of computation and communication. The following section describes different types of auctions.

3.3 Types of Auctions

Auctions play an important and recurring role in this thesis, since the simplified form of buyer-seller interaction they embody is closely related to several issues in our proposed framework. Both problems investigated in this thesis, task allocation and target tracking, address cases where multiple buyers and sellers are interacting. In this chapter, different types of simple auctions, and how they promote different kinds of behavior among agents are mentioned. The case of an agent auctioning one item to a set of agents is considered. Combinatorial auctions are also considered, in which the system agents are allowed to bid on bundle of items or goods. An underlying assumption made through the majority of this thesis is that each agent has a specific utility which determines its ability to handle the auctioned task. The auctioning mechanism is run to look for a better utility for the whole system.

3.3.1 Single-good Auctions

Since auctions are simply mechanisms for allocating goods, there are various types of auction that can achieve this goal. In the simplest familiar type of auctions, there is one good for sale, one seller, and multiple buyers. Each buyer has his own valuation for the good, and wishes to purchase it at the lowest possible price. These types of auctions are called single-sided auctions, as there are multiple single-sided agents on only one side of the auction. Thus, the main goal is to design a protocol for this auction that satisfies certain desirable global criteria, such as the expected revenue of the seller. It is not surprising that the literature on this topic is vast. Fig. 3.1 shows a classification taxonomy for the different types of auctions as proposed by Reck [85].

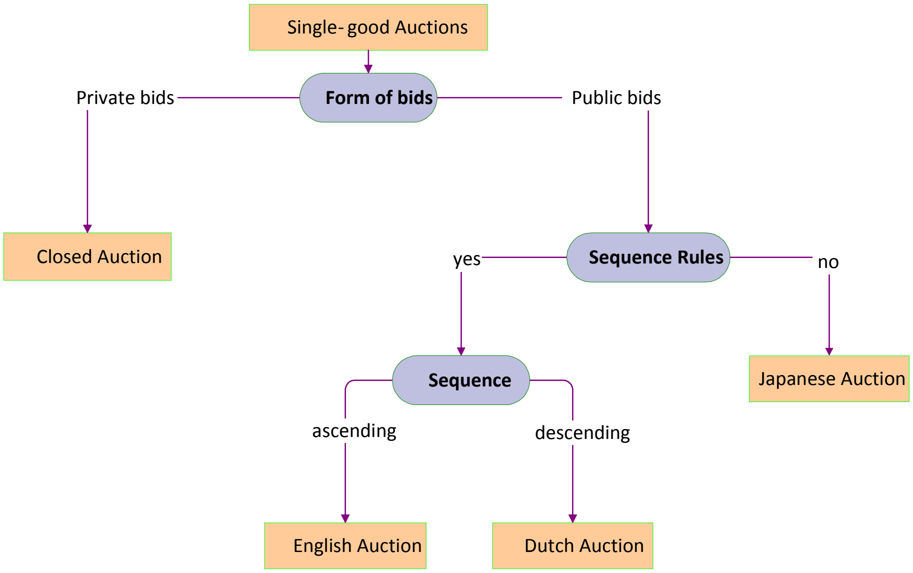


Figure 3.1: Auction taxonomy

As shown in Fig. 3.1, auctions are classified based on the form of bids, the sequencing rules and the ordering mechanism, and the mode of transaction occurrence. In this section a brief idea about these types of auctions is presented.

- **English auctions**: This type of auction is also called ascending-bid auctions [86]. It is perhaps the most well known type of auctions. In other words, when people speak of auctions, they usually mean this type. The auctioneer sets a starting price for the good or the service he has. Each buyer then announces a bid for this good or service. The buyers then increase their bids until one buyer wins this good or service after the auction closes. The rules for when the auction closes vary. In some instances, the auction closes after a specific period of time, in others it closes when no new bids are made for a period of time. The winner must purchase the announced good or service after the auction ends.

- **Japanese auctions**: These are simply a special form of English auction [83]. It has an ascending bid feature, as English auction. In other words, the agents have the option to increase their bids if they are interested in buying the announced good or service. It differs from the English auction in that each agent must choose whether or not he is willing to continue to bid or drop out the auction. The auctioneer calls out successively increasing prices in a regular fashion with the same restriction. Any agent who chooses to drop out of the auction can't reenter the auction. The auction ends when there is only one agent left.

- **Dutch auctions**: In a Dutch auction, the auctioneer starts at a very high price and then proceeds to announce successively lower prices in a regular fashion at specific time intervals. Thus, it is also called a descending bid-auction. The current price is displayed on a screen seen by all agents. The auction process continues until the first moment an agent signals the auctioneer by pressing a buzzer to stop the clock. The winning agent must then purchase the good for the displayed price.

- **Sealed-bid or closed auctions**: The aforementioned types of auctions are considered as open-bid auctions, as the buying agents announce their bids in public. However in the sealed-bid auction family [87], each agent submits a sealed bid to the auctioneer for the announced good. This sealed bid is not accessible to other agents. The agent with the highest bid must purchase the good. If the auction type is first price sealed bid, the winner agent pays a price, which is equal to their own bid, whereas in a second price sealed bid auction, the winner pays a price equal to the second highest bid. The second price sealed bid auction is also known as Vichrey auction.

Generally speaking, any framework structure for negotiation is considered to be an auction. Each negotiation structure has some bidding and clearing rules. These rules depend completely on the type of auction used.

3.3.2 Combinatorial Auctions

In this type of auctions, the agents are allowed to place bids on combinations (bundles) of items rather than bidding on each item separately. This is done under the constraint that one bundle can be allocated to no more than one agent. In the combinatorial auctions, the bidders' valuation for the bundle item is not the sum of their valuations. Combinatorial auctions are extensively studied by economists [87]. In many systems, the problem is that there is a set of tasks or jobs that needs to be distributed among the agents but the agents have complex preferences over the set of tasks. An example for that is a workflow application in which there are a set of workflows, each composed of a set of web services, and there are a number of agents responsible for completing these services [86]. These services must be performed by certain deadlines. Each agent can perform a subset of the services but each agent has different costs. The agent costs might depend on the agent's type, its current load, and the services it has performed before, etc. The goal is to allocate the workflows to agents so that the total number of completed workflows is maximized.

Recently, combinatorial auctions have been employed in many applications such as truckload transportation, bus routes, and industrial procurement, and airport arrival and departure slots, task allocation, as well as for allocating radio spectrum for wireless communications services [87]. All of the aforementioned single good auctions can be used as combinatorial auctions, but with auctioning a bundle of items instead of a single item.

3.3.2.1 Winner Determination Strategy

This subsection discusses winner determination algorithms for combinatorial auctions. Assume that the auctioneer has a set of items, $M = 1; 2; ...; m1$, to sell, and the buyers submit a set of bundle bids, $B = \{B_1; B_2; ...; B_n\}$. A bundle bid can be defined as $B_j = \{Se_j; p_j\}$, where $Se_j \in M$ is a set of items and $p_j > 0$ is a price. From the point of view of the auctioneer, the ultimate goal of its winner determination strategy is to find the optimal bids that maximize its revenue. For single-good auction mechanisms, the winner determination is a computationally simple task. However, the winner determination in the combinatorial auctions is a computationally hard task [88], as the auction setting allows complex bids such as bundle bids.

A. Search Formulation

In order to search for the best bids that increase the revenue of the auctioneer (winner determination problem), the submitted bids are structured in the form of a tree. After constructing the bids tree, standard search algorithms can be used to solve the winner determination problem. Two types of the search trees are explained in brief.

- **Search on items**: In this type of search [86, 89], the goal is to solve the winner determination problem using AI-search algorithms over all possible allocations. To do that, the submitted bids are organized in the form of a search tree. A path from the root to any leaf of the conducted tree corresponds to a set of bids, with a constraint that no two bids share an item. In other words, each path in the search tree consists of a sequence of disjoint bids. Before constructing the search tree, the received bids are listed. For each bid, the items are shown but the price is not. The first level of the tree is the root, and any other level is considered as the child of the root. The first step in constructing the tree starts from the top by drawing all the children that have item 1 in them. All bids that do not include items that are already used on the path are included on each node of the tree as the children of the current node. The order of the bids on a path does not affect the branching factor of the tree [89]. An example of this search tree is shown in Fig. 3.2.

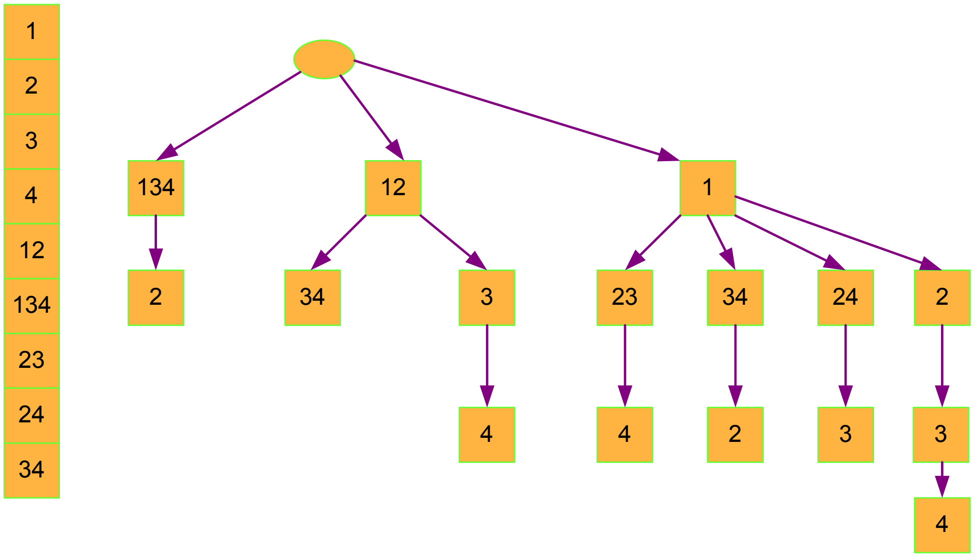


Figure 3.2: Search on items

The auctioneer's possibility of keeping items can be implemented by placing dummy bids of price zero on those items that received no 1-item bids [90]. This may happen when the auctioneer finds that its revenue will increase if some items are kept.

- **Search on bids**: Another form of the search trees is search on bids tree [86, 89]. Each node in this tree is a bid submitted by a bidder. Reaching any edge of the tree represents whether that particular bid is accepted as a solution or not. The children of a bid in this tree are the world where that bid is accepted (IN), and the world where that bid is rejected (OUT). An example of this search tree is shown in Fig. 3.3

  The black boxes in Fig. 3.3 indicate search paths that denote a complete set of bids. In other words, no more bids can be added because they contain items already sold. This type of search tree is considered to be a faster winner determination algorithm than the previous search on items type. The reason for this is that the bid order can

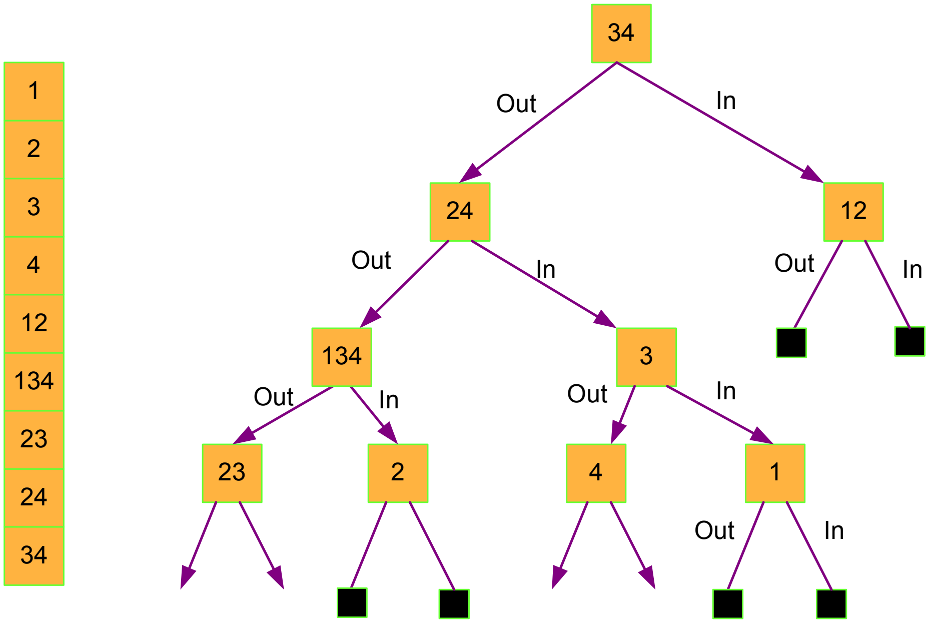


Figure 3.3: Search on bids

be optimized. Another advantage of this search tree over the search on items tree is that the addition of dummy singleton bids is not needed.

B. Search Strategy

In the previous subsection, two types of search trees are discussed. Different search algorithms such as depth first search (DFS), breadth first search (BFS), branch and bound, A* and iterative deepening A* (IDA*) [86] can be used to search theses trees. In this subsection, two search techniques are discussed: BFS, and DFS. These search techniques have been used in this thesis as explained in the next chapter.

- **Depth First Search (DFS)**: Depth first search is mainly proposed in many applications for traversing trees and graphs [83]. In this type of search, a random node is selected as a starting node to begin the search process. As illustrated in Fig. 3.4, the algorithm proceeds to the unvisited children of this node in a one-by-one fashion until reaching the end of the path. The search process then backtracks to do the same with another node until scanning the whole tree or graph.

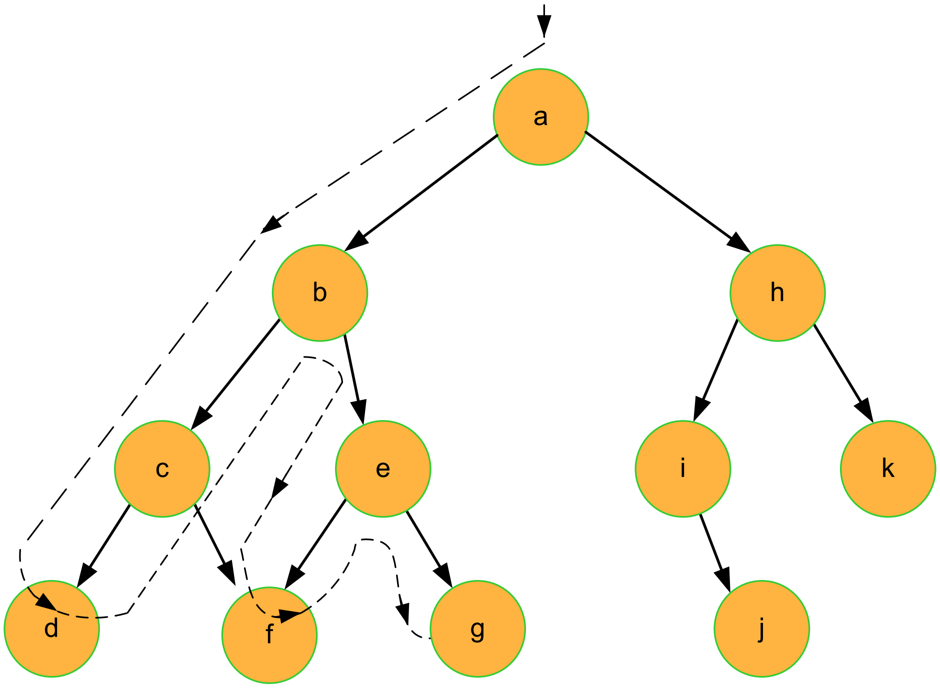


Figure 3.4: Depth first search

The main advantage of DFS is that only the nodes on one search path need to be kept in memory at any one time [89]. Another advantage of this algorithm is that a solution can exist at any time during the search process, and the quality of the solution improves until reaching the best solution at the end of the search process [89]. The main disadvantage of DFS is that it may get stuck exploring a long blind branch of the tree while there is a solution path of only one or two steps.

- **Breadth First Search (BFS)**: Breadth first search is another way of traversing trees and graphs [83]. As illustrated in Fig. 3.5, the algorithm starts from the root of the tree or graph and then proceeds to the neighboring nodes in one to one fashion. The adjacency list of each node is scanned at most once. Thus the BFS algorithm examines the whole tree without considering the goal solution until it finds it.

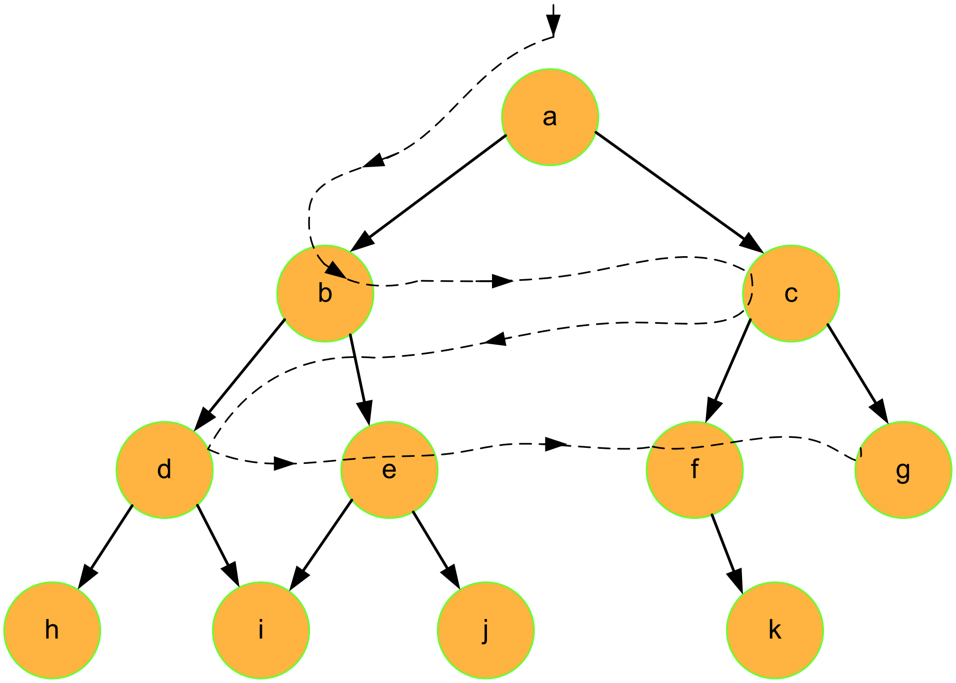


Figure 3.5: Breadth first search

The main advantage of the BFS algorithm is that it always finds the shortest path first or at least, the solution that has the least number of steps. BFS is very systematic and is guaranteed to find the solution if it is exists [83]. On the other hand, BFS may use more memory but will always find the shortest path first or at least, the solution that has the least number of steps.

C. Bidding Languages

As explained above, the agents can communicate their bids to the auctioneer. In combinatorial auctions, a bundle of items are available to the bidders to buy. The bidders must specify a valuation for every number of items in these types of auction. Bidders would be provided with some bidding languages that would allow them to convey their bids more compactly [83]. Bidding languages basically try to efficiently model different patterns for bids. The usefulness of a bidding language depends on the sorts of underlying valuations that bidders will commonly want to express. Examples for that are:

- **OR-bids**: In this case, agents can place multiple atomic bids for non-overlapping sets of items. They would be happy to accept any number of the bids they submit. Assuming an agent is placing bids on items AB, B, and C. The bidder will be willing to win AB OR ABC OR B OR BC OR C. The OR-bids bidding language doesn't add any computational burden on the bidder, but it has two main disadvantages. First, it is not always clear to the bidder what combinations of items he may win. Second, the bidder can't express substitutes like A or B but not both.

- **XOR-bids**: The agent can place multiple bids for sets of items and is willing to win any one of the bids, but not more than one combination of bids [86]. XOR-bids take the form of a series of atomic bids joined together by an exclusive-OR operation. If for example a bidder wants any combination of A, B, and C. This bidder must bid explicitly on each combination: ABC AB AC BC A B C. The main advantage of the XOR-bids bidding language is that all possible combinations of bids can be represented. For example, the bidder can use it to place a bid that says they are willing to pay $5 for item 1 or $5 for item 2 or $7 for both but want to buy at most one of them. It is also simple for a bidder to understand which combination of items he may win. The main problem in this bidding language is that the number of bids must be limited. Another practical problem with XOR bids is that most of the winner determination algorithms are designed to work with OR bids [86].

3.4 The Pros and Cons of Market-based Approaches

The decision to use auctions in this thesis comes from the existence of several desirable properties of auction approaches [18, 22]. These advantages can be summarized as follows:

- **Efficiency**: One of the greatest strengths of market approaches is their ability to utilize the local information and preferences of their participants to arrive at an efficient solution given limited resources [91]. Market-based approaches have elements that are centralized and distributed [91]. Thus they can produce efficient solutions by capturing the respective strengths of both distributed and centralized approaches. It has been shown in [91–93] that efficient solutions can be produced by market approaches with respect to a variety of team objective functions. As will be explained later in this thesis, our proposed framework contains these two types of elements.

- **Robustness**: Fully centralized approaches employ a single agent to coordinate the entire team in a multiagent system. They may suffer from a single point of failure, and have high communication demands. Market-based approaches do not require a central coordinator agent and therefore there is no common-mode failure point or vulnerability in the system [17, 19, 80]. These approaches can be made robust to several types of malfunctions, including complete or partial failures of agents [22].

- **Scalability**: As mentioned before, the computational and communication requirements of market-based approaches are usually manageable, and do not prohibit these systems from providing efficient solutions because they are not fully centralized systems. Thus, as the size of the inputs in the system increases, these approaches can still provide an efficient solution [22]. Market-based approaches can scale well in applications where the team mission can be decomposed into tasks that can be independently carried out by small sub-teams [91].

- **Online input**: Market-based approaches are able to seamlessly incorporate the introduction of new tasks [94]. Market-based approaches can often incorporate online tasks by auctioning new tasks as they are introduced to the system or generated by the agents themselves [91].

- **Uncertainty**: Market-based systems are able to operate in unknown and dynamic environments by allowing team members to adapt cost estimates over time, and reallocate tasks when appropriate [95].

Although market-based approaches have many advantages, they are not without their disadvantages. Perhaps the biggest drawback of market-based approaches is the lack of formalization in designing appropriate cost and revenue functions to capture design requirements [91]. Also, negotiation protocols, developing appropriate cost functions, and introducing relevant penalty schemes can complicate the design of the market approach [18]. In domains where fully centralized approaches are feasible, market-based approaches can be more complex to implement, and can produce poorer solutions [91]. Also, when fully distributed approaches suffice, market-approaches can be unnecessarily complex in design and can require excessive communication and computation [91]. Finally, incorporating contract breaches with appropriate penalties, developing more sophisticated methods for cooperative handling of partial malfunctions and repairs, and evaluating response speed and robustness to a variety of failures are still challenges needing to be tackled by researchers [91].

3.5 Concluding Remarks

In this chapter, the market-based approach was discussed. This approach imitates the auction process of buying and selling services through bidding and can provide an efficient way to coordinate the activities of a number of competitive agents. The auction process and different types of auctions have been highlighted in this chapter, focusing on combinatorial auctions. A number of key theoretical, practical, and computational insights of combinatorial auctions are explained in brief. The pros and cons of the market approaches have also been discussed.

Chapter 4

Task Allocation in Mobile Surveillance Systems

The problem of task allocation in mobile surveillance systems is a twofold one. First it addresses how to assign a set of tasks to a set of mobile sensors. Second it considers how to coordinate the behavior of the sensor team in order to do the cooperative tasks efficiently. This problem is usually studied as Multi-Robot Task Allocation (MRTA) problem [46]. In spite of the great number of MRTA algorithms reported in the literature, important aspects have, to date been given little attention. These aspects include but are not restricted to allocation of complex tasks, dynamic task allocation, and constrained task allocation. In this thesis, we are trying to address these aspects by giving a unified framework to study this problem in a formal manner. The thesis discusses centralized and hierarchical dynamic and fixed tree task allocation approaches to solve the MRTA problem

The remainder of the chapter is organized as follows. The related work is presented in section 4.1. Section 4.2 introduces the task allocation problem's definition and formulation. Section 4.3 discusses the different components of the proposed market-based approach. The conclusion and future work are summarized in section 4.5.

4.1 Related Work

MRTA approaches can be classified based on problem description, task allocation category, the planning method, the organizational paradigm used, and problem solving techniques. A framework for studying MRTA problem is introduced in [96]. According to this framework, the multi-robot task allocation problem can be seen as an instance of an optimal assignment problem (OAP) [97]. This can be defined in the following way: given n robots, and a m single robot tasks, assign robots to tasks so as to achieve maximum overall profit. Because the problem of task allocation is a dynamic decision problem that varies in time with phenomena including environmental changes, this static assignment problem should be solved iteratively over time [96]. Thus, dynamic task allocation is a class of task allocation in which the assignment of robots to sub-tasks is a dynamic process and may need to be continuously adjusted in response to changes in the task environment or group performance [20].

A formal analysis and taxonomy of multi-robot task allocation is also introduced in [98]. The authors in this survey paper tried to provide a particular taxonomy for studying MRTA, based on organizational theory from several fields, including operations research, economics, scheduling, network flows, and combinatorial optimization. Also, complete analyses and description for single-task (ST), multi-task (MT) robots, single robot (SR), multi-robot (MR) tasks, instantaneous assignment (IA) and time-extended assignment (TA) are provided. ST means that each robot is capable of executing at most one task at a time, while MT means that some robots can execute multiple tasks simultaneously. Similarly, SR means that each task requires exactly one robot to achieve it, while MR means that some tasks can require multiple robots. In IA approaches [46, 99, 100] the available information concerning the robots, the tasks, and the environment permits only an instantaneous allocation of tasks to robots (i.e., task independence is a strong assumption). These approaches are sometimes used in order to avoid the need for highly computationally scheduling algorithms. At the other extreme is continuous task allocation or time extended assignment (TA) approaches [18, 19, 101] where more information is available, such as the set of all tasks that will need to be assigned. Because robots have to reason about the dependencies between tasks, TA is more demanding from a planning perspective. The work presented here can be categorized as single robot task- single task robot- instantaneous assignment task allocation (ST-SR-IA).

Existing task allocation techniques can be categorized to: (1) Allocation of simple tasks, (2) Allocation of complex tasks. Simple tasks are tasks that can be accomplished in a straightforward manner [18, 46, 99–102] while complex tasks are the tasks that have several possible ways of implementation [19, 103, 104]. When dealing with complex tasks, the structure and semantics of the tasks can be exploited to produce more efficient team plans by giving individual robots the ability to come up with new ways to perform a task, or by allowing multiple robots (”mobile sensors” henceforth) to cooperate by sharing the subcomponents of a task, or both [19]. Motivated by the sparse attention given to formal modeling, and analysis of complex task allocation, complex tasks that can be decomposed into different subtasks is one of focuses of this thesis.

From the perspective of planning, there are two common approaches to the task allocation problem: decompose-then-allocate and allocate-then-decompose. In the first technique, the complex mission is decomposed to simple subtasks and then these subtasks are allocated to the team members based on their capability and availability to complete the subtasks as required [105, 106]. In this type of techniques, the cost of the final plan cannot be fully considered, because the task decomposition is done without knowing to whom tasks will be allocated. Another disadvantage of this type is inflexibility to changes in the designed plan. So, the plan designed by the central agent cannot be rectified even if it is found costly. On the other side, in the allocate-then-decompose approach [46], the complex tasks are allocated to mobile sensors, and then each mobile sensor decomposes the awarded tasks locally. The main disadvantage of this approach is the allocation of all tasks to only one mobile sensor and thus, the preferred task decomposition is purely dependent on the plan of that mobile sensor, which increases the possibility of reaching a suboptimal solution. It may be more beneficial to allocate tasks to more than one mobile sensor in order to consider different plans for the required task. While the decompose-then-allocate and the allocate-then-decompose methods may be capable of finding feasible plans, there are drawbacks to both approaches. Motivated by these drawbacks, Zlot and Scentz proposed in [19] a market-based task allocation approach to allocate complex tasks among a robot team. They proposed a solution concept that unifies the decompose-then-allocate and allocate-then-decompose stages by not decoupling the solution into separate allocation and decomposition phases.

Another line of comparison between task allocation approaches is the classification according to team organization: centralized and hierarchical approaches. In centralized approaches, a single agent is employed to coordinate the entire mobile sensor team. Theoretically, this agent gathers all relevant information from the team members, does planning for the entire team, and broadcasts commands in order to allocate tasks to robots. Practically, fully centralized approaches can be computationally intractable, brittle, and unresponsive to change. Thus, for applications where teams are small and the environment is static or global state information is easily available, centralized approaches are the best-suited solution. Not surprisingly, many MRTA architectures implement some form of this approach [55, 100, 107–109]. On the other hand, in hierarchical task allocation approaches, mobile sensors rely solely on local knowledge. Such approaches have many advantages over centralized approaches, such as flexibility, robustness, and low communication demands. However, because a good local solution may not sum to a good global solution, hierarchical approaches can produce highly suboptimal solutions. Fully hierarchical schemes are best suited in applications where large teams carry out relatively simple tasks without efficiency restriction. In order to gain the advantages of both schemes, many market-based approaches have been proposed [17, 21, 80]. Thus having centralized and hierarchical elements can help in accomplishing task allocation mission.

This chapter presents a market-based approach to complex task allocation for multisensor surveillance systems. Both centralized and hierarchical allocations are investigated as winner determination strategies for different levels of allocation and for static and dynamic search tree structures. Details of the proposed approach are presented in the next sections.

4.2 Complex Task Allocation

This section provides problem definition for both simple and complex task allocations and formulation for only the complex task allocation.

4.2.1 Problem Definition

DEFINITION 1: Simple Task Allocation
Given a set of mobile sensors S each looking for one task, and a set of tasks T each requires one sensor, the simple task allocation can be defined by a function $A : T \rightarrow S$, mapping each task to a mobile sensor in order to be executed.

DEFINITION 2: Complex Task Allocation
Given a set of mobile sensors S, and a set of tasks T, let $G \subset T$ be a group or a bundle of tasks that is decomposable into other tasks $Md \in G$. The complex task allocation can be defined by a function $Bf : Md \rightarrow S$, mapping each subtask to a mobile sensor to be responsible for completing it.

For both simple and complex task allocation, the goal is to assign sensors to tasks so as to maximize overall expected performance, taking into account the priorities of the tasks and the skill ratings of the sensors. Appropriate functions are needed to map possible task outcomes into revenue values and to map possible schemes for performing the task into cost values. The goal is to assign tasks to sensorss such that the overall profit (the excess of revenue over cost) is maximized. Generally, mobile sensors receive revenue and incur costs for accomplishing a specific team-task. A mobile sensor can also receive revenue from another sensor in exchange for goods or services. The price dictates the payment amount for the good or service. A common approach is to bid for a good or service in order to arrive at a mutually acceptable price [18].

4.2.2 Problem Formulation

The problem of task allocation can be formulated in many ways. Given our surveillance application domain, it can be formulated as follows:

1. AOI: two dimensional, bounded area of interest.
2. S: a team of mobile sensors $s_i, i = 1, 2, ...n$. It is assumed that each sensor carries sensors (such as cameras, sonar and laser range finders)
3. T: a set of tasks $t_j, j = 1, 2, ...nt$.
4. U: a set of sensors utilities, u_{ij} is the utility of sensor i to execute task j.

For a single sensor task, the problem is to find the optimal allocation of sensors to tasks, which will be a set of sensor and task pairs [45]:

$$(s_1, t_1), (s_2, t_2), ...(s_k, t_k) \; for 1 \leq k \geq m \tag{4.1}$$

For the general case, the problem is to find the optimal allocation of a set of tasks to a subset of sensors, which will be responsible for accomplishing it [19]:

$$Af : T \rightarrow S \tag{4.2}$$

Each mobile sensor $s \in S$ can express its ability to execute a task $t \in T$, or a bundle of tasks $G \subseteq T$ through bids $b_s(t)$ or $b_s(G)$. The cost of a bundle of tasks can be simply computed as the sum of costs of the individual tasks:

$$b_s(G) = \sum_{k=1}^{f} b_s(t_k) \{t_k \in G\} \tag{4.3}$$

where f is the number of tasks of the bundle G . The group's assignment determines the bundle $G \subseteq T$ of tasks that each mobile sensor $s \in S$ receives. These bundles can be characterized as follows:

$$\beta = \left\{(G_1, G_2, ...G_w) | G_{k_1} \cap G_{k_2} = \varphi, \bigcup G_w = T\right\} \tag{4.4}$$

The global objective function can vary depending on the requirements of the system or the preferences of the designer. The most common global objective is to minimize the sum of the team member costs, which can be described mathematically as follows:

$$C(A) = \sum_{s=1}^{n} b_s(G_s) \tag{4.5}$$

where $C(A)$ is the total required cost for executing the allocation A, and G_s is the bundle of tasks that is won by sensor s.

Though the mobile sensor team members may have well-defined cost or utility functions, these functions still rely on having accurate models of the world state and may require computationally expensive operations. When there are multiple goal locations like in surveillance application, determining the cost to perform even one task can require solving multiple path planning problems. Thus an instance of the traveling salesman problem

(TSP) [110, 111] might be used. In the theory of computational complexity, the decision version of TSP belongs to the class of NP-complete problems. Thus, it is assumed that there is no optimal algorithm for solving traveling salesman problems. In this work, a shortest sequence planning algorithm (SSP) [112] is used in order to find the minimum cost path for each mobile sensor given the tasks' locations. In this algorithm, an agent is tasked with visiting a set of points with the goal of finding out in which order it should visit these points so as to minimum traveling distance without going back to its original place, thus accomplishing the required tasks with near optimal system performance.

4.3 Proposed Market-based Approach

Market-based approaches have received significant attention and are growing very rapidly in the last few decades, especially in multi-agent domains [17, 19, 21, 79, 80]. These approaches are considered to be hybrid approaches that combine the centralized and distributed strategies (i.e., market-based approaches have elements that are centralized and distributed). Motivated by this regular attention, a market-based approach for dynamic task allocation for multisensor surveillance systems is presented in this chapter.

4.3.1 Single-shot and Combinatorial Auctioning

So far, researchers have studied single-item auctions at which items are auctioned off one at a time [113]. However, if there are strong synergies between the items of the bidders, highly suboptimal team solutions can result from single-item auctions [84]. Two items are said to exhibit positive or negative synergy for a bidder if the combined bid of this bidder on these two items is larger or smaller than the sum of its individual bids on each item separately.

An example of that is shown in Fig. 4.1. There is positive synergy between AOI-1 and station AOI-2 for the mobile sensor S_1 because they are close to each other. The mobile sensor S_1 can reach AOI-2 with a short distance (5 m) after it has reached AOI-1 (5 m). So, the sum of the single bids of S_1 on AOI-1 and AOI-2 (12=5+7) is more than the combined bid of S_1 on both areas (10=5+5). On the other hand, there is a negative synergy between AOI-3 and AOI-1 for S_1 because they are on opposite sides of the S_1, and hence the mobile sensor S_1 can therefore reach either one of the areas only with a long travel distance after it has reached the other one.

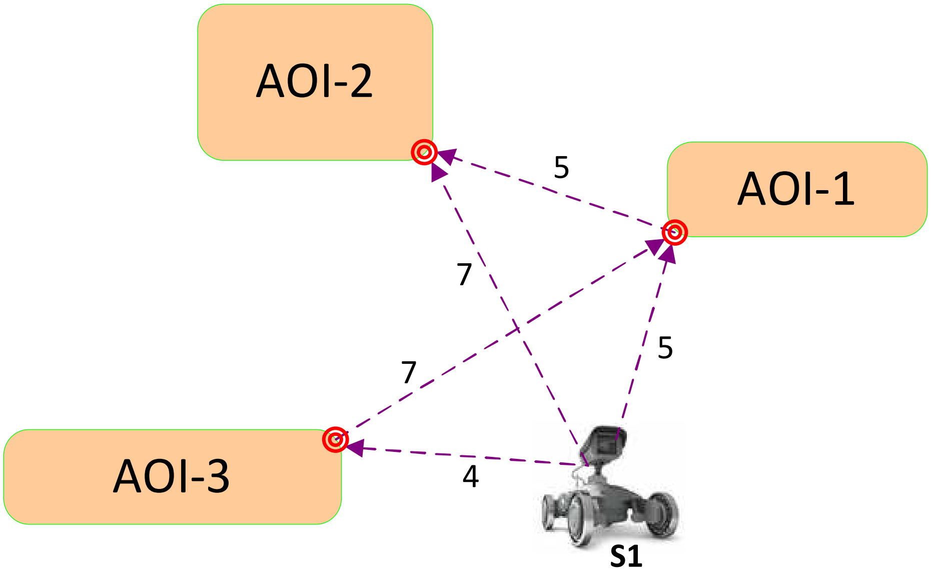


Figure 4.1: Single-shot and combinatorial auctioning

Generally speaking, combinatorial auctions attempt to overcome the disadvantages of single-item auctions by allowing bidders to bid on bundles of items [19, 84]. If a bidder wins a bundle, they win all the items in that bundle.

4.3.2 Auction Design

The task allocation approach proposed in this thesis imitates the auction process of buying and selling services through bidding. Sellers or auctioneers are responsible for processing the bids sent by buyers or bidders, and for determining the winning bidder. In this subsection, a brief idea about how to design the auction process will be based on maximizing a utility function.

4.3.2.1 Utility Function

Utility is the quality or state of being useful. For the task allocation problem, utility is a satisfaction (value of profit) derived by a mobile sensor s from accomplishing a task t.

Given a mobile sensor s and a task t, if s is capable of executing t, the utility a mobile sensor s from accomplishing a task t can be defined [19] on some standardized scale as:

$$u_{st} = p_s(t) - d_s(t) \tag{4.6}$$

Where $p_s(t)$ is the total payment it receives after executing the task t, and $d_s(t)$ is the total distance it travels to reach the task. The priorities of tasks to be executed should be taken into account while designing the task allocation framework. Our objective is to find the optimal assignment of tasks T to sensors S in order to minimize cost and thus maximize the overall utility. Consequently, system performance is ideally optimized. Thus, the goal is to assign sensors to tasks so as to maximize the cost, as we assume that there is no payement received after executing the task.

4.3.2.2 Search Tree

Most of task allocation approaches have treated tasks as atomic units [18, 46, 99, 101], thus allowing only static descriptions for each task, and so the only degree of freedom is determining to which sensor the task will be assigned. While this description is fine in case of simple tasks, it is not with complex tasks. Given the bid submitted, searching over all possible allocations can be used as a winner determination strategy. In this case, a search tree can be used as a better description for the tasks. In this tree, mobile sensor team members are permitted to bid on nodes representing varying levels of task abstraction, thereby enabling hierarchical planning, task allocation, and optimization among the team members.

In our work, the complex tasks to be allocated are structured as an ordered tree. In a set theory, a tree is defined to be a set E and a relation F where $F \subseteq E$ such that:

- F is a partial-ordering of E.
- For any $e \in E$,$\{$ $r \in E$ — $rFe\}$ is well-ordering.

The nodes (elements of the tree) that are immediately greater than a node are called its children, while the node that is immediately less is its parent (if it exists). Any node less is an ancestor and any node greater is a descendant. A node with no ancestors is a root. The partial ordering represents distance from the root, and the well-ordering requirement prohibits any loops or splits below a node (that is, each node has at most one parent, and therefore at most one grand-parent, and so on). In other words, if rFe then there is exactly one kFe such that rFk and there is nothing between r and k. Perhaps the best way to illustrate the mechanics of the task tree is through a simple example. Fig. 4.2 shows a surveillance scenario, which represents a shopping mall in the city of Waterloo, Ontario, Canada. The mission is to monitor a set of areas of interest (AOIs) such Zellers, Old Navy, the Bay, Zehrs, Winners and Sports Check. For the small areas like (Old Navy, Winners and Sports Check), only one or two vantage points are enough to achieve the monitoring task, while three points are required to cover the large areas like (Zellers, the Bay, and Zehrs).

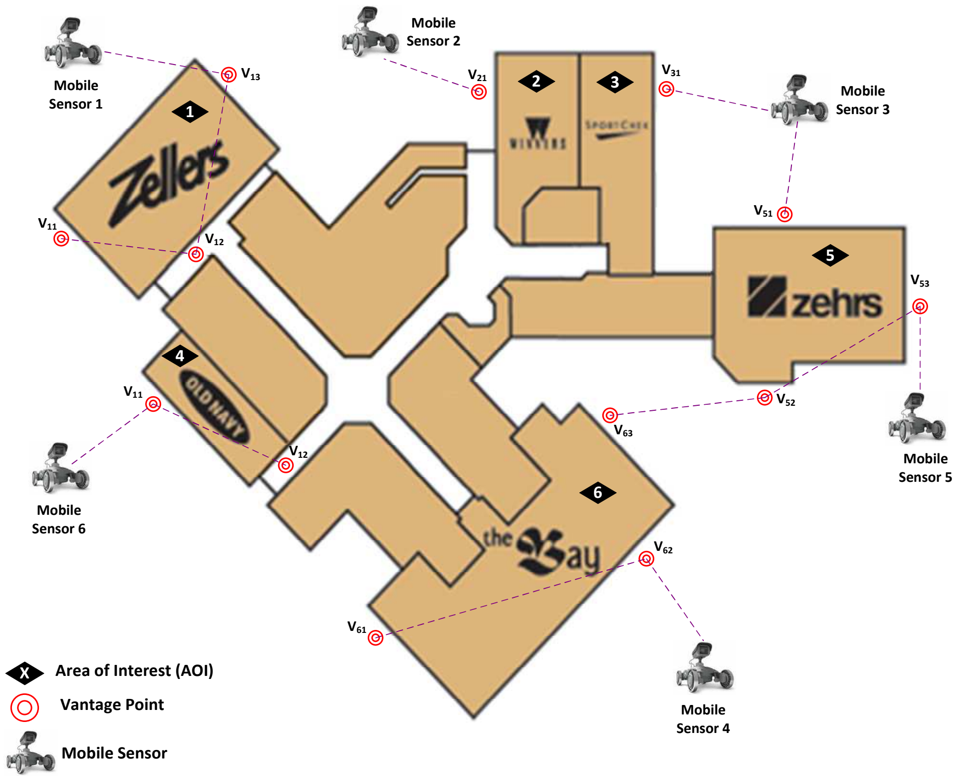


Figure 4.2: Surveillance scenario

The *AND/OR* task tree is constructed by decomposing the surveillance mission as a complex task into two subtasks (scan large areas and scan small areas) as illustrated in Fig. 4.3. Accomplishing the requested mission requires achieving scans of both large and small areas. In other words, these two subtasks are related to each other by the logical operator *AND*, which means that both tasks are required to be executed. The subtask (Scan large areas) is in turn decomposed to other simpler tasks such as Scan AOI-1, AOI-5 and AOI-6. The simple tasks can be executed by one of two plans, which contain the most primitive tasks. For example, to scan AOI-1, Plan-1 or Plan-2 can be chosen. These two alternative covering plans which are related to each other by the logical operator *OR* are computed based on the minimum traveling distance and the second minimum traveling distance. Plan-1 contains a list of primitive tasks (Goto V_{13}, Goto V_{12}, Goto V_{11}) that must be executed sequentially. Similarly, Plan-2 contains the same primitive task, but with different order.

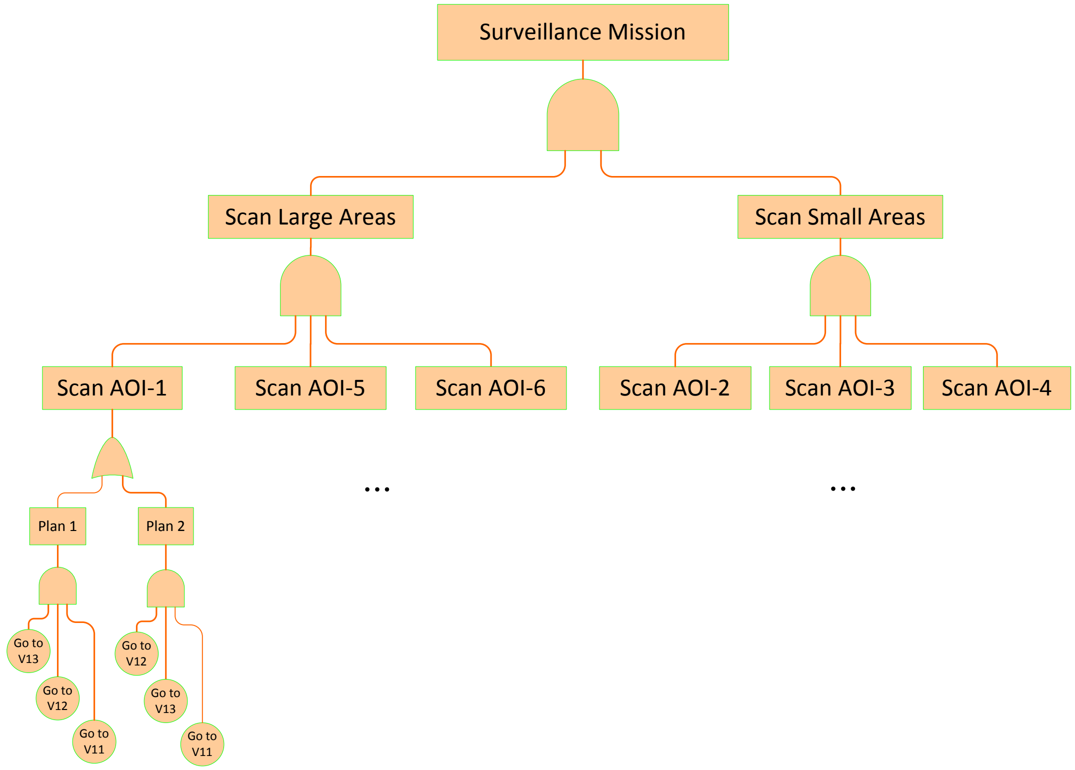


Figure 4.3: *AND/OR* task tree

This decomposition is done initially by an operator or by the selected initial auctioneer. Once the task tree is constructed and the decomposition is complete, the auctioneer holds a task tree auction, distributing tasks among the team and allowing other robots to use their own plans when appropriate. The auctions then proceed in rounds in which each mobile sensor holds a task tree auction (if it has any tasks) in a round-robin fashion.

Other logical operators like *XOR*, and *NAND* can also be used. The *XOR* operator can be used in order to implement the surveillance of sensitive areas (a sensitive area must be surveyed exclusively by the assigned sensor), while *NAND* can be used to prevent redundancy (i.e., each area of interest is surveyed by only one sensor).

4.3.3 Allocation Levels

From the perspective of planning, there are different allocation levels for complex tasks. In this subsection, these allocation levels are discussed in detail.

4.3.3.1 Point-Level Allocation

The complex mission is decomposed to simple subtasks and then these subtasks are allocated to the team member based on their capability and availability to complete the subtasks as required [105, 106]. In this type of techniques, the cost of the final plan cannot be fully considered because the task decomposition is done without knowing to whom tasks will be allocated. Another disadvantage is inflexibility to changes in the designed plan. So, the plan designed by the central agent cannot be rectified even if it is found costly. This is also called the decompose-then-allocate approach [104].

As shown in Fig. 4.4, the mission is initially decomposed by the auctioneer into a set of surveillance points. All auctions are only for tasks in this set of goal points. In other words, there is no notion of abstract tasks, and no further decomposition occurs. Each winner puts the awarded subtask into its schedule to be considered for execution.

4.3.3.2 Area-Level Allocation

In this type of allocation techniques, the situation is different. The complex tasks are allocated to one mobile sensor, which in turn decomposes the awarded tasks locally. This is also called allocate-then-decompose approach [45]. One disadvantage of this approach is that it may be beneficial to allocate subcomponents of these tasks to more than one sensor. As shown in Fig. 4.5, the mission is allocated to one of the sensors (auctioneer), which in turn will be responsible for decomposing it to a set of surveillance areas. All auctions are only for tasks in this set of goal areas. In other words, there is no notion of single tasks; no

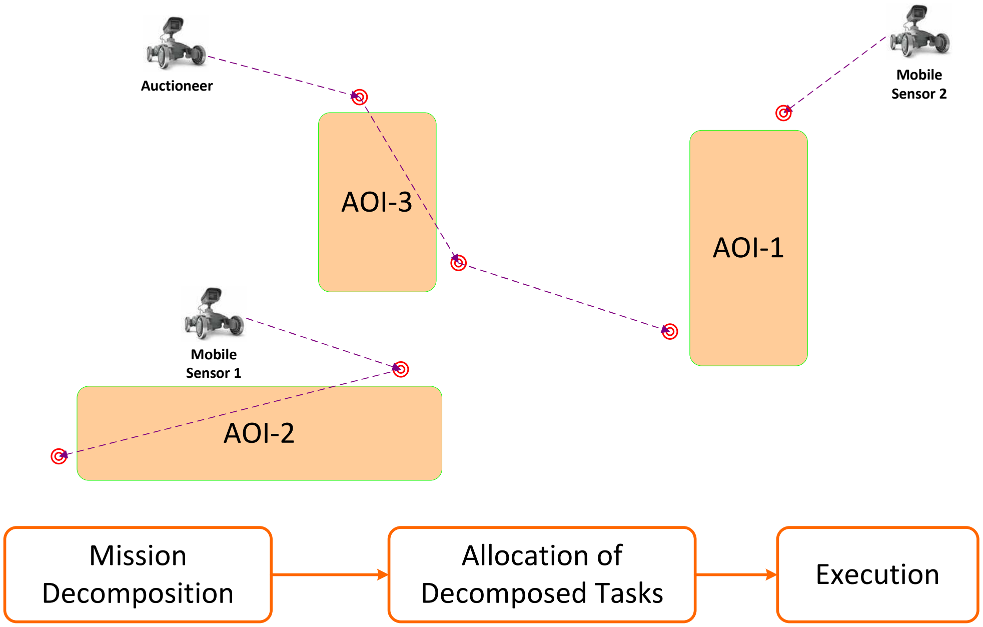


Figure 4.4: Decompose-then-allocate approach

further decomposition occurs. Each winner puts the awarded subtask area into its schedule to be considered for execution. The auctioneer finds itself the best to execute area 3, while it awards area 1 to mobile sensor 2, and area 2 to mobile sensor 1.

4.3.3.3 Mission-Level Allocation

In this type of allocation, the auction is for the whole mission and so it is considered as a single-shot type of auctioning. Thus the entire mission is awarded to only one sensor, which can decompose the mission to subtasks as shown in Fig. 4.6. The decomposition is considered only for execution, not for reallocating the subtasks. This can be called a higher-level allocate-then-decompose [19]. The main disadvantage of this approach is the allocation of all tasks to only one sensor and thus, the preferred task decomposition is purely dependent on the plan of that sensor, which increases the possibility of using a suboptimal solution. It may be more beneficial to allocate tasks to more than one sensor

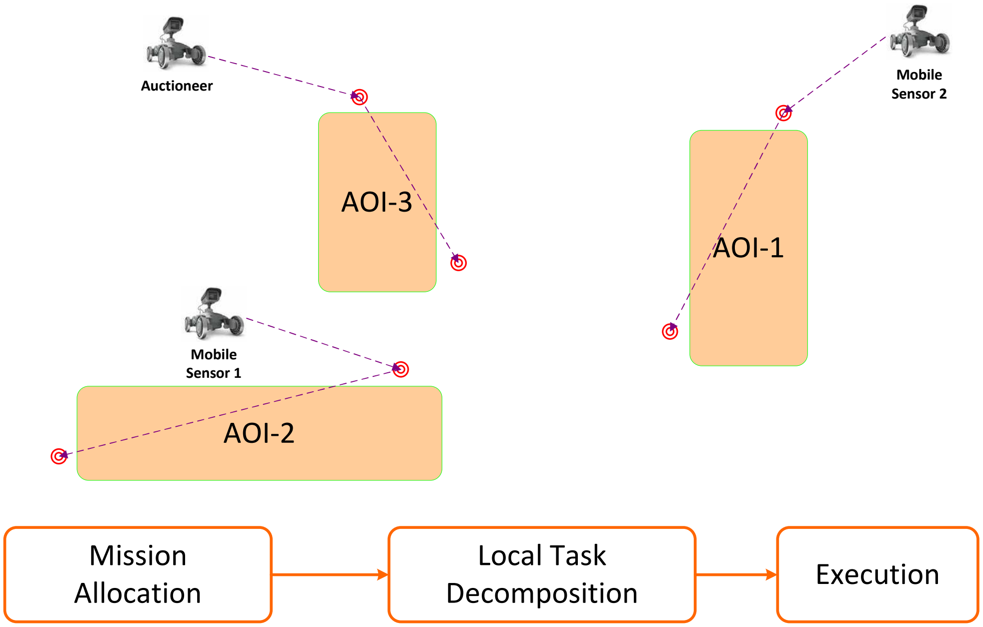


Figure 4.5: Allocate-then-decompose approach

in order to consider different plans for the required task.

4.3.4 Winner Determination Strategies

The winner determination strategy addresses how to optimally find the set of bids that maximize the bidder's revenue. In a combinatorial auction, winner determination is an *NP*-hard problem [114], as searching for all possible allocations of items to mobile sensors is computationally intractable, and no approach will work in polynomial time.

Winner determination strategy is highly affected by the type of description of tasks to be allocated. As mentioned previously, the complex tasks to be allocated are represented as an ordered tree. Breadth and depth first search algorithms are used to find the task allocation solution from this task tree structure. Two organizational paradigms, namely, centralized and hierarchical allocation, are used during the allocation process. These paradigms

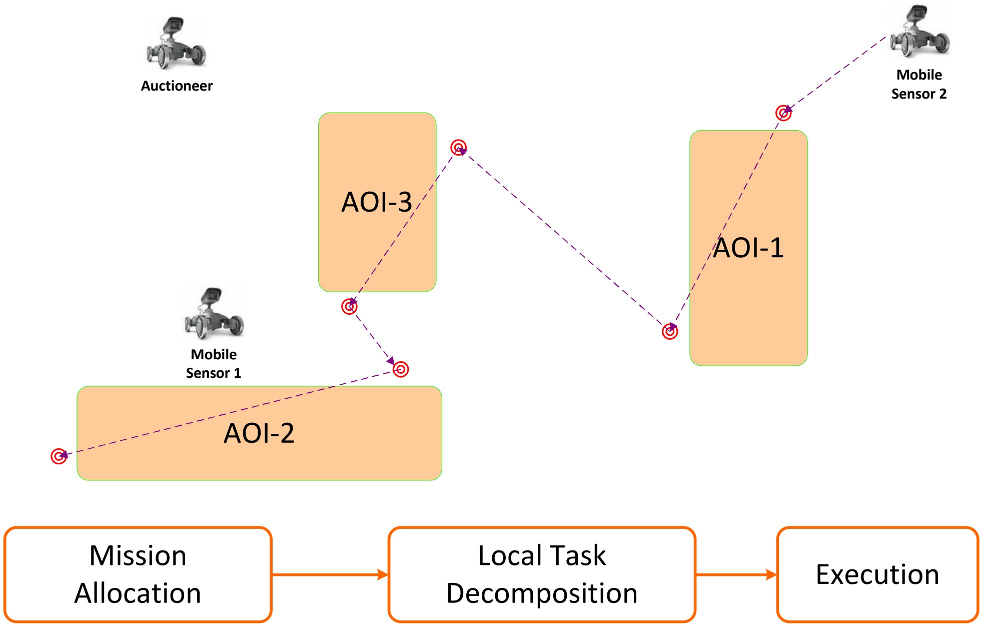


Figure 4.6: Higher-level allocate-then-decompose approach

determine the roles, relationships, and structures which govern the auction process.

4.3.4.1 Centralized Allocation

In this type of auctioning, an auctioneer holds a series of auctions to allocate the surveillance tasks to the mobile sensors in order to maximize the system utility. An example of this is shown in Fig. 4.7. The auctioneer holds auctions in rounds to allocate the tasks it has to the mobile sensors S_1, S_2, S_3, and S_4, if it finds that the system utility will increase.

4.3.4.2 Hierarchical Allocation

As shown in Fig. 4.8, the tasks are allocated initially to the mobile sensors S_1, S_2, S_3, and S_4 via a central auctioneer. Each mobile sensor can hold auctions in rounds for the tasks it wins in the initial auction.

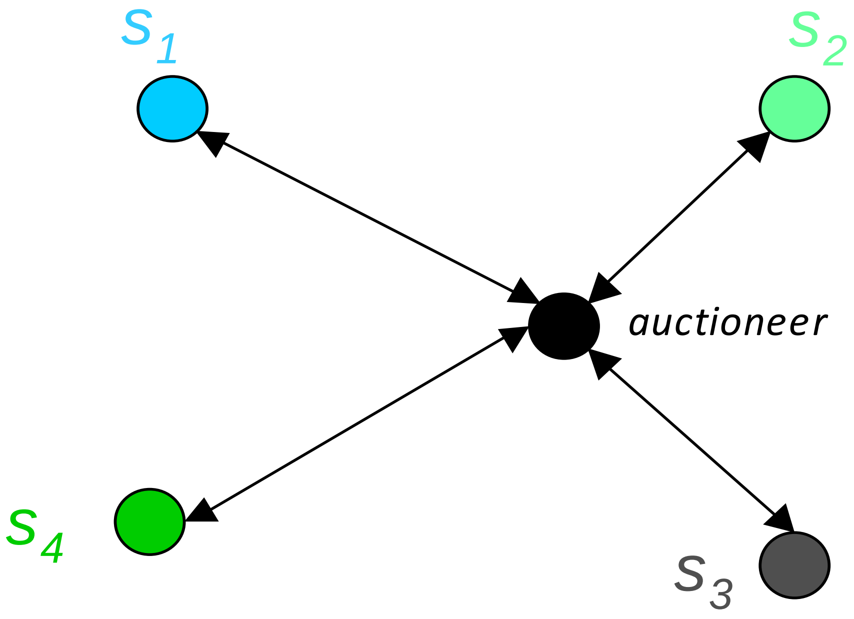


Figure 4.7: Centralized auctioning

4.4 Fixed and Dynamic Tree Allocation

The key to effective task allocation for multi-sensor systems is to iterate the assignment, in order to deal with changes in the tasks, the sensors, and the environment [80]. For that, an iterated market-based complex task allocation approach is developed to allocate tasks to the sensor team members through contract negotiation. A manager sensor can offer tasks to other sensors, which may submit bids based on their ability to perform the tasks. Centralized and hierarchical auction mechanisms are developed for complex task allocation. Using fixed and dynamic tree are explained in the following subsections.

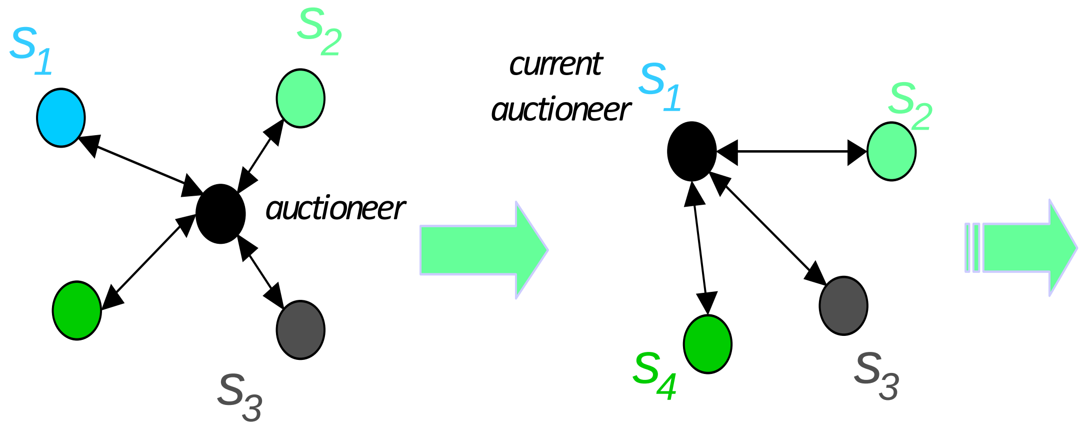


Figure 4.8: Hierarchical auctioning

4.4.1 Fixed Tree Task Allocation

Consider a team of mobile sensors assembled to perform a particular task. Consider further, that each mobile sensor is capable of executing one task at once, and each task can be accomplished by one sensor. The task information is continuously available to the mobile sensors team. Thus, the proposed approach in this case can be framed as iterated instances of ST-SR-IA (Single-Task Single-Robot Instantaneous-Assignment). The goal of the team is to perform the task efficiently while minimizing costs. This can be done by modeling each mobile sensor as self-interested agent which aims to minimize its individual cost and so maximize the whole team's performance by following a greedy algorithm. Each mobile sensor is either cooperating with other members of the team to achieve an outcome greater than that possible by each member alone, or competing with other members to perform the required task at the lowest possible cost, thus eliminating waste and inefficiency. A system such as this can be widely-used in the economy, and so many desirable characteristics from the market mechanisms might be used.

In the market-based task allocation approaches [18, 19, 96], the mission task to be executed is awarded to one of the sensors, which is called the operator or the auctioneer. This auctioneer is responsible for providing a plan to the other team members for executing this mission task. The proposed plan by the auctioneer is implemented in the form of a task tree. Each mobile sensor maintains several lists of trees it has agreed to handle. The constructed task tree will have abstracted nodes (have children) and primitive nodes (no children). The proposed algorithm allows the auctioneer to sell any primitive node through iterated single-shot auctions. Also, combinatorial-auctions are adopted by selling some bundles of tasks or abstracted nodes. Our algorithm begins by auctioning the abstracted nodes in the constructed tree from the top to the bottom of the tree. Selling the topmost abstracted or apex node in the tree means that one sensor will be responsible for executing the whole mission task. This is of course if its execution cost is less than the execution cost of the auctioneer. If after auctioning all the abstracted nodes, there are still nodes in the auctioneer tree. It tries to sell them by running auctions on the level of the primitive nodes, if it is profitable for it to do so. The winner of any auction will insert the node it wins into its task tree and thus be responsible for completing it either by executing it itself, or by selling the whole task or part of it to other teammates.

Generally, the breadth first search is a graph search algorithm that begins at the root node and explores all the neighboring nodes. Then, for each of these neighboring nodes, the search algorithm explores their unexplored neighbor nodes, and so on, until it finds the goal solution. In this thesis, a top-down BF search is used to decide the order of selling the tree nodes. In other words, the proposed BF fixed tree algorithms starts by selling the abstract nodes and then selling the single tasks at the lower levels which were not sold while selling the abstract nodes. From the perespective of bidding language, as explained above, only *AND* bidding language means that the current auctioneer will have only one plan for each area it has, whereas for the *AND/OR* bidding language, the current auctioneer will have some abstract areas that must to be surveyed (*AND*) with two alternative plans for each area, and only one will be sold. The details of the centralized proposed breadth first

(BF) fixed tree allocation for only AND and for AND/OR bidding languages are shown in algorithms 1 and 2 respectively.

Algorithm 1: CenBFFixedTree1Plan($T, Aucid, Ns, Na$) Centralized Breadth First Fixed Tree (Only AND)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$:the cost of executing the tree $T$
begin
    Compute the task tree $T$ using $SSP$ (shortest sequence planning) algorithm; the
    number of abstract nodes is $N$;
    Initialization: $Aucid \longleftarrow T$; mark the whole tree as won by $Aucid$;
    for $i = 1$ to $N$ do
        Compute $cost_1(T)$; the traveling cost of the tree $T$ before auctionning ;
        for $j = 1$ to $Ns$ do
            while $j \neq Aucid$ do
                Compute $b_j(i) = \sum b_j(h) \{h \in i\}$; the bid of sensor $j$ on node $i$ ;
        Find $bestsensor_1$; the $ID$ of mobile sensor which has the best bid on node $i$.
        This is done using $SSP$ algorithm ;
        $T_{bestsensor1} \longleftarrow T_{bestsensor1} + i$;
        $T_{Aucid} \longleftarrow T_{Aucid} - i$;
        Compute $cost_2(T)$; using $SSP$ algorithm ;
        if $cost_2(T) > cost_1(T)$ then
            $T_{bestsensor1} \longleftarrow T_{bestsensor1} - i$;
            $T_{Aucid} \longleftarrow T_{Aucid} + i$;
    if $\exists$ set $t \subset T_{Aucid}$ then
        for $c = 1$ to $t$ do
            Compute $cost_3(T)$; the traveling cost of the tree $T$ before auctionning the
            single nodes;
            for $o = 1$ to $Ns$ do
                while $o \neq Aucid$ do
                    Compute $b_o(c)$; the bid of sensor $o$ on node $c$ ;
            Find $bestsensor_2$; the $ID$ of mobile sensor which has the best bid on node
            $c$. This is done using $SSP$ algorithm ;
            $T_{bestsensor_2} \longleftarrow t_{bestsensor2} + c$;
            $T_{Aucid} \longleftarrow T_{Aucid} - c$;
            Compute $cost_4(T)$; the traveling cost of the tree $T$ after auctionning;
            if $cost_4(T) > cost_3(T)$ then
                $T_{bestsensor2} \longleftarrow T_{bestsensor2} - c$;
                $T_{Aucid} \longleftarrow T_{Aucid} + c$;
    $Totalcost \longleftarrow cost(T)$;
```

Algorithm 2: CenBFFixedTree2Plans($T, Aucid, Ns, Na$) Centralized Breadth First Fixed Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
      $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFFixedTree1Plan($T_1, Aucid, Ns, Number$);
    Compute $cost(T_1)$; the traveling cost of the tree $T_1$ ;

    Compute the second best tree $T_2$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_2$; mark the whole tree as won by $Aucid$;
    CenBFFixedTree1Plan($T_2, Aucid, Ns, Number$);
    Compute $cost(T_2)$; the traveling cost of the tree $T_2$ ;
    if $cost(T_2) > cost(T_1)$ then
        $T \longleftarrow T_1$;
    else
        $T \longleftarrow T_2$;
    $Totalcost \longleftarrow cost(T)$;
```

In the context of a fixed task tree allocation, a set of constraints dictates that the whole auction mechanism is based only on one task tree, which can be proposed by the operator or the auctioneer. The proposed algorithms allow using only one auctioneer from the start to the end of auctioning, and so are considered as a centralized task allocation as shown in algorithms 1, and 2 . It also allows changing the auctioneer during auctioning while considering only the plan of the original operator. In this case, the proposed mechanism can be seen as a hierarchal task allocation mechanism. Algorithms 3, and 4 show BF hierarchical fixed tree allocation for only AND and for AND/OR bidding languages. Another constraint which is considered in all proposed algorithms dictates that at most one node can be sold to each bidder per auction. This is because upon awarding one node to a bidder the bid prices on other nodes become invalid due to the fact that bid prices are conditioned on the current commitments of each participant.

Algorithm 3: HBFFixedTree1Plan($T, Aucid, Ns, Na$) Hierarichical Breadth First Fixed Tree (Only AND)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
      $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFFixedTree1Plan($T, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFFixedTree1Plan($T, i, Ns, Number$);
    $Totalcost \longleftarrow cost(T)$;
```

Algorithm 4: HBFFixedTree2Plans($T, Aucid, Ns, Na$) Hierarichical Breadth First Fixed Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFFixedTree1Plan($T_1, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFFixedTree1Plan($T_1, i, Ns, Number$);
    Compute $cost(T_1)$; the traveling cost of the tree $T_1$ ;
    Compute the best task tree $T_2$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_2$; mark the whole tree as won by $Aucid$ ;
    CenBFFixedTree1Plan($T_2, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFFixedTree1Plan($T_2, i, Ns, Number$);
    Compute $cost(T_2)$; the traveling cost of the tree $T_2$ ;
    if $cost(T_2) > cost(T_1)$ then
        $T \longleftarrow T_1$;
    else
        $T \longleftarrow T_2$;
    $Totalcost \longleftarrow cost(T)$;
```

On the other hand, a depth first (DF) search is an algorithm for traversing or searching a tree, or a graph, starting at a selected node and exploring as far as possible along each branch before doing the same with neighboring nodes. The proposed task allocation mechanism uses DF search as an alternative way to decide the order of selling the tree nodes. Thus, the proposed DF fixed tree algorithms tries to find what is the best way to sell the tree nodes, either by bidding the single tasks of each abstract node separately, or by bidding the abstract node as a bundle of single tasks. The details of the proposed centralized DF fixed tree allocation for only *AND* and for *AND/OR* bidding languages are shown in algorithms 5 and 6 respectively, whereas the hierarchical DF fixed tree allocation for *AND/OR* bidding languages is shown in algorithm 7.

Algorithm 5: CenDFFixedTree1Plan $(T, T_2, T_3, Aucid, Ns, Na)$ Centralized Depth First Fixed Tree (Only AND)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$:the cost of executing the tree $T$
begin
    Compute the task tree $T$ using $SSP$ path planning algorithm; the number of abstract nodes is $N$;
    Initialization: $Aucid \longleftarrow T$; mark the whole tree as won by $Aucid$;
    for $i = 1$ to $N$ do
        Compute $cost_1(T)$; the traveling cost of the tree $T$ before auctionning ;
        $Copy(T, T_1)$; copy the tree $T$ in $T1$;
        Find $Nc = Children(i)$; the number of children of node $i$;
        for $c = 1$ to $Nc$ do
            for $j = 1$ to $Ns$ do
                while $j \neq Aucid$ do
                    Compute $b_j(c)$; the bid of robot $j$ on node $k$ ;
            Find $bestsensor_1$; the $ID$ of the sensor which has the best bid on node $c$. This is done using $SSP$ algorithm ;
            $T_{bestsensor1} \longleftarrow T_{bestsensor1} + c$;
        $T_{Aucid} \longleftarrow T_{Aucid} - i$;
        Compute $cost_2(T)$; the traveling cost of the tree $T$ after auctionning ;
        $Copy(T, T_2)$; hold this solution ; Compute $cost_2(T)$; the traveling cost of the tree $T$ after auctionning ;
        $Copy(T_1, T)$; retrieve the old tree;
        for $j = 1$ to $Ns$ do
            while $j \neq Aucid$ do
                Compute $b_j(i) = \sum b_j(M) \{M \in i\}$; the bid of sensor $j$ on node $i$ ;
        Find $bestsensor_2$; the $ID$ of the sensor which has the best bid on node $i$. This is done using $SSP$ algorithm ;
        $T_{bestsensor_2} \longleftarrow T_{bestsensor_2} + i$;
        $T_{Aucid} \longleftarrow T_{Aucid} - i$;
        $Copy(T, T_3)$; hold this solution;
        Compute $cost_3(T)$; the traveling cost of the tree $T$ after auctionning ;
    $mincost = min(cost_1(T), cost_2(T), cost_2(T))$;
    if $mincost = cost_1(T)$ then
        $Copy(T_1, T)$; retrieve the first solution;
    if $mincost = cost_2(T)$ then
        $Copy(T_2, T)$; retrieve the second solution;
    if $mincost = cost_3(T)$ then
        $Copy(T_3, T)$; retrieve the third solution;
    $Totalcost \longleftarrow cost(T)$
```

Algorithm 6: CenDFFixedTree2Plans $(T, T_5, T_6, Aucid, Ns, Na)$ Centralized Depth First Fixed Tree (AND/OR)

```
Data: $Na$: = number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$:the cost of executing the tree $T$
begin
    Find the best task tree $T_5$ using $SSP$ path planning algorithm; the number of
    abstract nodes is $N$;
    Find the second best task tree $T_6$;
    Initialization: $Aucid \longleftarrow T_5$; mark the whole tree as won by $Aucid$;
    Compute $cost_1(T_5)$; the traveling cost of the tree $T_5$ before auctionning ;
    CenDFFixedTree1Plan $(T_5, T_1, T_2, Aucid, Ns, Na)$;
    Compute $cost_2(T_1)$;
    Compute $cost_3(T_2)$;
    Initialization: $Aucid \longleftarrow T_6$; mark the whole tree as won by $Aucid$;
    CenDFFixedTree1Plan $(T_6, T_3, T_4, Aucid, Ns, Na)$;
    Compute $cost_4(T_3)$;
    Compute $cost_5(T_4)$;
    $mincost = min(cost_1(T_5), cost_2(T_1), cost_3(T_2), cost_4(T_3), cost_5(T_4))$;
    if $mincost = cost_1(T_5)$ then
        $Copy(T_5, T)$; get the final solution;
    if $mincost = cost_2(T_1)$ then
        $Copy(T_1, T)$;
    if $mincost = cost_3(T_2)$ then
        $Copy(T_2, T)$;
    if $mincost = cost_4(T_3)$ then
        $Copy(T_3, T)$;
    if $mincost = cost_5(T_4)$ then
        $Copy(T_4, T)$;
    $Totalcost \longleftarrow cost(T)$
```

Algorithm 7: HDFFixedTree2Plans ($T, Aucid, Ns, Na$) Hierarichical Depth First Fixed Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
      $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Compute the second best task tree $T_2$ using $SSP$ path planning algorithm;
    CenDFFixedTree2Plans($T, T_1, T_2, Aucid, Ns, Na$); find the initial solution;
    $Copy(T, T_5)$; hold this solution;
    for $i$ to $Ns$ do
        CenDFFixedTree1Plan ($T, T_3, T_4, Aucid, Ns, Na$);
    $Totalcost \longleftarrow cost(T)$;
```

4.4.2 Dynamic Tree Task Allocation

The proposed fixed task tree allocation described in the previous section could be seen as an instance of the decompose-then-allocate approach. The main drawback of this approach is that the cost of the final plan cannot be fully considered because the complex task is decomposed by the auctioneer without knowledge of the eventual task allocation. Also, backtracking is not allowed in this approach, and so any costly mistakes in the auctioneer decompositions cannot be rectified. Generally, the allocate-then-decompose method tries to avoid the drawbacks of the decompose-then-allocate method. However, there are still some disadvantages. Motivated by the drawbacks of both methods, a dynamic tree allocation is proposed in this thesis to allow backtracking in order to recover the bad plans made by the auctioneers. The proposed dynamic tree mechanism allows auctioning on all levels of abstraction of the mission task implemented by the task tree from the top to the bottom. Each mobile sensor evaluates its ability to execute the required task based on its plan, not on the plan of the auctioneer. The proposed dynamic tree allocation algorithms are either executed by allowing only one auctioneer (centralized allocation), or by allowing different auctioneers (hierarchical allocation).

The details of proposed centralized BF dynamic tree allocation for only *AND* and for *AND/OR* bidding languages are shown in algorithms 8 and 9 respectively, whereas, the hierarchical DF dynamic tree allocation for only *AND* and for *AND/OR* bidding languages are shown in algorithms 10, and 11. Also, the details of the hierarichical DF dynamic tree allocation for *AND/OR* is shown in algorithm 12.

Algorithm 8: CenBFDynamicTree1Plan $(T, Aucid, Ns, Na)$ Centralized Breadth First Dynamic Tree (Only *AND*)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
      $Totalcost$: the cost of executing the tree $T$
begin
    Compute the task tree $T$ using $SSP$ path planning algorithm; the number of
    abstract nodes is $N$;
    Initialization: $Aucid \longleftarrow T$; mark the whole tree as won by $Aucid$;
    for $i = 1$ to $N$ do
        $\forall i$ (abstract node), find the area index $A(i)$;
        Compute $cost_1(T)$; the traveling cost of the tree $T$ before auctionning ;
        for $j = 1$ to $Ns$ do
            while $j \neq Aucid$ do
                Compute $P_j(A(i))$; the plan of mobile sensor $j$ for area $A(i)$;
        Find $bestsensor_1$; the $ID$ of the sensor which has the best plan on node $i$. This
        is done using $SSP$ algorithm ;
        $T_{bestsensor1} \longleftarrow T_{bestsensor1} + P_{bestsensor1}(A(i))$;
        $T_{Aucid} \longleftarrow T_{Aucid} - i$;
        Compute $cost_2(T)$; the traveling cost of the tree $T$ after auctionning ;
        if $cost_2(T) > cost_1(T)$ then
            $T_{bestsensor1} \longleftarrow T_{bestsensor1} - P_{bestsensor1}(A(i))$;
            $T_{Aucid} \longleftarrow T_{Aucid} + i$;
    if $\exists$ set $t \subset T_{Aucid}$ then
        for $c = 1$ to $t$ do
            Compute $cost_3(T)$; the traveling cost of the tree $T$ before auctionning the
            single nodes;
            for $o = 1$ to $Nr$ do
                while $o \neq Aucid$ do
                    Compute $b_o(c)$; the bid of the sensor $o$ on node $c$ ;
            Find $bestsensor_2$; the $ID$ of the sensor which has the best bid on node $c$.
            This is done using $SSP$ algorithm ;
            $T_{bestsensor_2} \longleftarrow t_{bestsensor2} + c$;
            $T_{Aucid} \longleftarrow T_{Aucid} - c$;
            Compute $cost_4(T)$; the traveling cost of the tree $T$ after auctionning ;
            if $cost_4(T) > cost_3(T)$ then
                $T_{bestsensor2} \longleftarrow T_{bestsensor2} - c$;
                $T_{Aucid} \longleftarrow T_{Aucid} + c$;
    $Totalcost \longleftarrow cost(T)$
```

Algorithm 9: CenBFDynamicTree2Plans $(T, Aucid, Ns, Na)$ Centralized Breadth First Dynaamic Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFDynamicTree1Plan($T_1, Aucid, Ns, Number$);
    Compute $cost(T_1)$; the traveling cost of the tree $T_1$ ;

    Compute the second best tree $T_2$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_2$; mark the whole tree as won by $Aucid$;
    CenBFDynamicTree1Plan($T_2, Aucid, Ns, Number$);
    Compute $cost(T_2)$; the traveling cost of the tree $T_2$ ;
    if $cost(T_2) > cost(T_1)$ then
        $T \longleftarrow T_1$;
    else
        $T \longleftarrow T_2$;
    $Totalcost \longleftarrow cost(T)$;
```

Algorithm 10: HBFDynamicTree1Plan($T, Aucid, Ns, Na$) Hierarichical Breadth First Dynamic Tree (Only AND)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFDynamicTree1Plan($T, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFDynamicTree1Plan($T, i, Ns, Number$);
    $Totalcost \longleftarrow cost(T)$;
```

Algorithm 11: HBFDynamicTree2Plans $(T, Aucid, Ns, Na)$ Hierarichical Breadth First Dynamic Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
      $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_1$; mark the whole tree as won by $Aucid$ ;
    CenBFDynamicTree1Plan($T_1, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFDynamicTree1Plan($T_1, i, Ns, Number$);
    Compute $cost(T_1)$; the traveling cost of the tree $T_1$ ;
    Compute the best task tree $T_2$ using $SSP$ path planning algorithm;
    Initialization: $Aucid \longleftarrow T_2$; mark the whole tree as won by $Aucid$ ;
    CenBFDynamicTree1Plan($T_2, Aucid, Ns, Number$);
    for $i = 1$ to $Ns$ do
        CenBFDynamicTree1Plan($T_2, i, Ns, Number$);
    Find $cost(T_2)$; the traveling cost of the tree $T_2$ ;
    if $cost(T_2) > cost(T_1)$ then
        $T \longleftarrow T_1$;
    else
        $T \longleftarrow T_2$;
    $Totalcost \longleftarrow cost(T)$;
```

Algorithm 12: HDFDynamicTree2Plans $(T, Aucid, Ns, Na)$ Hierarichical Depth First Dynamic Tree (AND/OR)

```
Data: $Na$: number of areas;
      $Ns$: number of mobile sensors;
      $Aucid$: the $ID$ of the auctioneer
Result: $A : S \longleftarrow T$; the allocation of the task tree $T$ to the set of mobile sensors $S$;
        $Totalcost$: the cost of executing the tree $T$
begin
    Compute the best task tree $T_1$ using $SSP$ path planning algorithm;
    Compute the second best task tree $T_2$ using $SSP$ path planning algorithm;
    CenDFFixedTree2Plans($T, T_1, T_2, Aucid, Ns, Na$); find the intial solution;
    $Copy(T, T_3)$; hold this solution;
    for $i = 1$ to $Ns$ do
        Find $T_1(i)$; find the second best tree for each robot;
        CenDFFixedTree2Plans ($T, T_3, T_2(i), Aucid, Ns, Na$); find the final solution;
    $Totalcost \longleftarrow cost(T)$;
```

4.5 Conclusion

In this chapter, a market-based approach has been presented to solve the multisensor task allocation problem in mobile surveillance systems. Centralized and hierarchical algorithms with fixed and dynamic trees have been examined focusing on complex tasks, i.e., the tasks that can be decomposed into subtasks. Both breadth first, and depth first search mechanisms have been used for deciding the order of bidding. As shown in chapter 6, the results of the conducted experiments show that hierarchical dynamic tree task allocation outperforms other existing techniques, especially in complex surveillance operations where a large number of mobile sensors is used to scan a large number of areas.

Chapter 5

Target Detection and Tracking

The main objective of this chapter is to give a detailed description of the developed methodology for tracking multiple objects, which will be incorporated into the proposed mobile surveillance framework. It is divided into five major sections. Section 5.1 gives a brief introduction to the target tracking problem and its description. Section 5.2 gives a detailed investigation of the target tracking problem from different dimensions. The state of the proposed work is determined in each dimension. The most recent related work, along with a brief discussion of the important problems that are observed in the design of a target tracking mobile sensor network, are given in section 5.3. The target tracking problem statement and formulation are introduced in sections 5.4, and 5.5 respectively. This is followed in section 5.6 by presenting the proposed target tracking approach. Finally, important conclusions are drawn in section 5.7.

5.1 Introduction

Recognizing the motion of objects navigating in a bounded area of interest is an essential task in a number of applications, such as security, surveillance, and reconnaissance. An important research issue that arises in these applications is target tracking: estimating the state of a target, based on some measurements from some static sensors, to maintain this target in view. Many factors make fixed sensor placement in advance inadequate for surveillance and security applications. These factors include the lack of prior information about the area of interest. Also, large areas of interest need a huge number of fixed sensors to accomplish the target-tracking mission.

Mobile sensors can be very effective in overcoming the disadvantages of fixed sensors [115]. The movement ability of the mobile sensors gives them the ability to achieve more thorough coverage of the area of interest than the combined coverage capabilities of these sensors. Although mobile sesors are good options for surveillance and security applications, there are also several challenges. First, there must be a coordination mechanism among the sensor team to coordinate the sensing operations. Without coordination, performance of the mobile sensors team is expected to be significantly degraded. Second, the mobile sensor must be deployed in the environment in such a way as to achieve full coverage of the environment, to do the required tasks with minimum energy consumption, and thus to minimize the cost of sensor deployment.

This chapter focuses on the development of an energy efficient target tracking methodology which is able to accomplish the tracking mission cooperatively. This is done by coordinating the mobile sensors using a market-based approach. The proposed target tracking methodology will be incorporated into the mobile surveillance framework.

5.2 A Classification Taxonomy of Current Target Tracking Approaches

Recently, various research groups have studied the target tracking problem from different perspectives. In [51], a taxonomy that classifies research in the field of target tracking according to the various problem definitions and evaluation criteria is presented. According to this taxonomy, the target tracking problem can be classified along a number of dimensions as follows:

5.2.1 The Number of Trackers versus the Number of Targets

The first classification axis for target tracking problem is the number of trackers, which can be divided into 'single' or 'multiple' based on whether cooperation among trackers is planned or not. In this thesis, any tracking technique without cooperation between trackers is considered a single tracker problem. The second axis is the number of targets, which affects the complexity of the problem. A tracking problem can be classified according to the combination of these two axes as:

Single Tracker Single Target (STST) The simplest form of this category is tracking of a single target by a single tracker. For the tracking system to perform properly, the most likely measured target location should be used to update the target's state estimator. This is generally known as the data association problem. The probability of the given measurement being correct is a distance function between the predicted state of the target and the measured state. A Probabilistic Data Association(PDA) filter is proposed in [116] to achieve successful tracking in cluttered environments. Also, the Kalman filter [117], and the particle filter [118] have been used to track a single moving target.

Single Tracker Multiple Targets (STMT) In this category, multiple targets are tracked by a single tracker. Multiple Hypothesis Tracking (MMT) , and joint probabilistic data association (JPDA) are the most well-known techniques in this category. A multiple hypothesis tracking algorithm is applied successfully for tracking multiple targets in [119]. Also, a decoupling joint probabilistic data association (DJPDA)algorithm is presented in [120] for multiple target tracking. This is done by decoupling the JPDA algorithm into separate probabilistic data association (PDA) algorithms.

Multiple Trackers Single Target (MTST) The main idea in this category is to track a single target with multiple trackers in order to improve the accuracy of target estimation. In other words, the measurements from several trackers are combined to decrease the measurement uncertainty [121]. This is very useful in some applications that require high accuracy such as ballistic missile defense, satellite defense, and air traffic control systems. A virtual Target Tracking (VTT) algorithm that can be applied to mobile satellite communication networks is proposed in [122].

Multiple Trackers Multiple Targets (MTMT) A lot of research in this category has been developed to track multiple targets with multiple trackers [49, 123]. The main focus of this category is to coordinate multiple trackers in order to track multiple targets efficiently and reliably.

In this thesis, a solution for the MTMT problem when trackers are mobile sensors is introduced. This focuses on the case when the number of targets is bigger than the number of trackers, which means that the cooperation between the trackers is indispensable.

5.2.2 Tracker Mobility

The tracking problem can be classified based on the number of degrees of possible tracker motions, such as stationary, pan/tilt/zoom, planar, and unrestricted.

Stationary: Most work in this category focuses on reliable perception because there is no motion control. Single or multiple stationary cameras are used as stationary trackers in [124].

Pan/Tilt: The use of autonomous pan-tilt cameras as opposed to static cameras can dramatically enhance the range and effectiveness of surveillance systems [125]

Planar: The tracker motion is planar. The design and implementation of an algorithm for landing a helicopter autonomously on a moving target using a mounted camera is a good example of this class [126]

Unrestricted: In this category, there is no restriction on the tracker motion. A good example of this is a camera mounted on a mobile sensor [49, 123].

The target tracking algorithms described in this thesis fall on the forth category.

5.2.3 Environment Complexity

The environment complexity is another factor to consider while tackling the target tracking problem. According to this, the environment complexity can be divided to:

Outdoor environment: In this type of environments, the main focus of the target tracking algorithms is on interactions between targets and trackers [49, 127].

Indoor environment: An example of this category is public places, such as airports. In this category, the tracking system should have the ability to use a path planning algorithm to track the targets in these envrionments [128].

The cooperative tracking algorithms presented in this thesis as part of the developed mobile surveillance framework are implemented in both outdoor and indoor environments.

5.2.4 Type of Cooperation

Cooperation between trackers is essential in MTST and MTMT problems to improve tracking performance. There are two different reasons for this cooperation:

Uncertainty reduction: Measurements from multiple trackers are combined to increase the accuracy of tracking (i.e., reduce the uncertainty). The measurement uncertainty due to clutters and missed detections in wireless sensor networks is addressed in [129] to track a single target.

Target allocation: The performance of tracking systems can be improved by allocating each group of targets to a single tracker in the best position. This type of cooperation is challenging, and has been addressed by many people [49, 123].

In this thesis, an energy efficient control algorithm for mobile sensors is developed to track moving targets according to the distribution of the targets in an environment. The proposed algorithm is an example of the second type of cooperation.

5.2.5 Coordination of Multiple Trackers

In the case of tracking multiple targets using multiple trackers, the effect of cooperation can be maximized by designing a suitable coordination strategy [51]. This coordination strategy modifies the behavior of the tracker either directly or by other trackers:

Explicit: The behavior of the tracker can be modified by other trackers using explicit communication [130]. An example of this; a tracker can stop another tracker from tracking a specific target if it thinks it is most suitable for the tracking system to exchange roles.

Implicit: Based on their best knowledge acquired from exchanging information or observing others, the trackers make their own decisions independently [131].

In this thesis, the mobile sensors share tracking information by broadcasting them. Each sensor does the tracking task independently but the target assignment is done by another senosr in the tracking system as will be seen later. Also, cueing and handoff for the exit targets is implemented through direct communication. These characteristics of the proposed framework put the work in this thesis under both categories.

5.2.6 Evaluation Criteria

In order to design tracking algorithms to be incorporated into the design of the multisensor surveillance framework, there should be an evaluation criterion to evaluate the framework performance. Out of many evaluation criteria, the following have been found important while designing our framework:

Tracking accuracy: Tracking accuracy is the most popular evaluation criterion taken into account in the literature, especially when tracking a single target.

Travel distance: The travel distance is one of the evaluation criteria that should be taken into account for evaluation of the tracking system. This occurs when the tracker is a mobile sensor.

Energy consumption: Since energy is the most limited resource in the tracking system, the designed tracking algorithm should be energy efficient to conserve this resource as much as possible. In [132], energy consumption is taken into account in transmitting and receiving messages for cooperation.

Speed of convergence: When tracking targets, speed of convergence is an interesting evaluation criterion. This occurs when the tracker is a mobile sensor.

In this thesis, the purpose of the single target tracking algorithm is to track positions of a moving target using a camera and laser rangefinder, so that the tracking accuracy and speed of convergence are used to evaluate the performance of the algorithm. On the other hand, in multi-target tracking, the goal is to keep tracking the moving targets with a minimum number of mobile sensors, so the energy consumption and travel distance are used to evaluate the performance of the algorithm.

5.3 Related Work

One of the fundamental issues that arise in a mobile sensor network is coverage. Coverage can be maximized by determining the optimal placement of static sensors as seen in the class of art gallery problems [133]. However, recent investigations reveal that mobile sensors can self-organize to provide better coverage than static sensors [115, 134]. In other words, mobile sensors are often desirable, since they can patrol a wide area and can be repositioned for better coverage [135]. A distributed and scalable potential field-based approach for the deployment of mobile sensors is introduced in [136] in order to achieve full coverage. The fields are constructed such that each sensor is repelled by both obstacles and by other sensors, thereby forcing the network to spread itself through the environment. In [137], algorithms for controlling a group of sensing vehicles to achieve full coverage in a dynamic environment are proposed. Nevertheless, collision avoidance and the effect of measurement errors can't be guaranteed. In [115], the authors proposed an autonomic mobile sensor architecture using an ant-like scheme to self-organize nodes to track unknown time-varying target distributions. Hoe and Varchney [134] proposed energy efficient deployment algorithms for a mobile sensor network. The main goal is to organize and optimize the network resources in terms of coverage, uniformity, and time and distance traveled, until the algorithm converges on a solution.

In [138], the problem of computing robot motion strategies that maintain visibility of a moving target in a cluttered workspace is introduced. The general problem is divided into two categories, on the basis of whether the target is predictable or not. For the predictable case, an algorithm that is based on numerical solutions is presented. Also, two on-line algorithms are presented to maintain future visibility with limited or partially-predictable target. Also in [139], the related question of maintaining the visibility of a moving target in a cluttered workspace by a single robot is addressed. A visibility region is associated with each robot, and the goal is o keep each target tracked by at least one robot.

In the area of multi-sensor surveillance, a coordinated multiple security sensor control system for warehouse surveillance and inventory assessment has been developed in [140]. For this goal, a supervised robotic security and inventory assessment system basically runs itself, until an unusual condition is encountered that requires human intervention. In other words, the developed system is semi-autonomous which utilizes autonomous navigation with human supervisory control when needed. Regarding the multi-target tracking needed in multisensor surveillance systems, an algorithm for computing the trajectories of multiple targets is developed in [141]. Also, a real time visual surveillance system for detecting and tracking multiple people and monitoring their activities in an outdoor environment is presented in [142]. The proposed system employs a combination of shape analysis and tracking to locate people and their parts (head, hands, feet, torso) and to create models of people's appearance.

Generally speaking, The recent interest in surveillance in public, military, and commercial scenarios is increasing the need to autonomously observe the movements of targets navigating in a bounded area of interest. This problem has been tackled by many researchers in the last few decades using different types of sensors. In [143], the problem of target tracking is tackled using stationary sensors. A single sensor and multisensor target tracking frameworks are proposed using extended Kalman filter. However, using mobile sensors is more beneficial than using stationary sensors for many reasons [123]. For example, there may be little prior information on the location of the area to be observed, which enforce the use of mobile sensors. Also, the targets to be tracked may exist in a large area which requires the placement of a large number of stationary sensors and so increases the system cost. Thus, the above constraints force the use of multiple sensors dynamically moving over time.

In [123], Parker investigated the use of a cooperative team of autonomous mobile robots for applications in the domains of surveillance, and reconnaissance. The main focus of this work is on developing distributed strategies that allow the team to attempt to minimize the total time in which targets escape observation by some mobile robot team member in the area of interest. The proposed strategies investigated the power of a weighted-force vector approach distributed across robot team members in simple, uncluttered environments that are either obstacle-free or have a random distribution of simple convex obstacles. In other work [131], the implementation of this weighted-force vector approach for fault-tolerant multi-robot cooperation is presented. The proposed approach is called A-CMOMMT (cooperative multi-robot observation of multiple moving targets). The main disadvantage of this approach is that one target can be observed by two robots because the algorithm is based on the density of targets in the vicinity of robots. Also, all robots are always exploring the environment which makes it unlikely to be an energy-efficient approach.

Kolling and Carpin [49] presented another version of the Parker work called B-CMOMMT to overcome some of the difficulties which exist in Parker work. One of these situations was when one robot followed two targets moving in opposite directions, eventually losing both. The loss of both targets may happen even when another robot is close by. The other robot is repelled from the first robot due to the force vector from that robot and has no chance of recapturing any of the targets. Intuitively the second robot should have attempted to capture one of the two targets instead of wandering around and attempting to discover an unknown target. B-CMOMMT operates under similar assumptions as those in the problem definition of A-CMOMMT, with the difference that it accommodates for varying restrictions of the sensors. An important difference is that in B-CMOMMT weights for robots forces are also considered, which rescales the robot forces' contributions. A robot can be in one of three modes (follow, explore, and help), and broadcasts help calls to indicate future target loss. A help call is broadcasted when target loss of a currently observed target is predicted. There is no guarantee that the robot in the best location will answer the help call. Also, the robot is always exploring the environment and, while exploring, checking unanswered help calls. If it doesn't find any, it will check the answered ones and the distances of the answering robots to the help calls. It then picks the closest help call for which it is closer than the answering robot.

Boyoon Jung [51, 144] introduced a target tracking problem using multiple, environment-embedded stationary and mobile sensors. In this work, an architecture for robot motion coordination, which exploits a shared topological map of the environment is presented. The environment is divided into regions, and so the author assumed that the topological map of the environment is already known. The stationary and mobile sensors maintain region-based density estimates for both the number of targets and the number of robots. According to these densities, the mobile robots move from less urgent regions to more urgent ones. The proposed algorithm is based on exploring the regions of the environment by the robots even if the regions are empty. Also, the possibility of tracking a group of targets by two robots is very high, so the proposed work is not an energy efficient one.

As seen in the literature survey above, the problem of multi target tracking which is tackled as part of the design of multi sensor surveillance systems has been solved using stationary sensors, mobile sensor, and a combination of both. Although this problem has been investigated thoroughly, most research in this area has focused on the development of a single accurate tracker (i.e, no cooperation between trackers). However, in security or surveillance applications, it is often more important to coordinate multiple sensors in order to accomplish the required mission efficiently and reliably. The existing multi target tracking approaches such as [49, 123, 131] don't consider the optimization of energy consumption during tracking. Also, in [51, 144], the proposed approaches are based on exploring the assigned region of the environment by the robots even if the regions are empty, which makes them unlikely to be energy-efficient approaches.

In this section, a review of current research work in the field of target tracking has been given. Three main issues can be drawn from this review:

1. The task of target tracking is challenging. This task can be studied from different points of view, such as the number of trackers, the number of targets, the mobility of trackers, the type of cooperation between trackers, etc.

2. The average accuracy of most target tracking techniques, especially in the multi-tracking ones, is not high. It is questionable whether such techniques can be used in

real-time target tracking systems.

3. It also seems that most of the current body of research in the field of target tracking focuses on studying the performance of target tracking techniques from various angles, but without much consideration of minimization of energy consumption.

In conclusion, the problem of multi-target tracking is challenging with many difficulties. Motivated by these difficulties, and apart from [49], an energy-efficient tracking algorithm to track multiple moving targets is presented in this thesis. The main goal is to achieve an improvement in the B-CMOMMT proposed in [49]. This can be done by guaranteeing that every help call is answered by only one mobile sensor, specifically, the closest sensor to the source of this help call. A market-based cuing/handoff approach is implemented to guarantee that the best suited sensor answers the help call. Each mobile sensor will have three modes of operation (follow, help, and sleep). The proposed approach in this thesis is good, when considered in terms of energy consumption.

5.4 General Problem Statement

One of the objectives of this thesis is to develop a generalized methodology for tracking of multiple moving objects. This methodology will be energy-efficient in such a way as to use the minimum number of resources to maintain continuous tracking of moving targets.

An autonomous surveillance system requires knowledge of the current targets' positions as well as their future positions, in order to be able to determine the sensor assignments and states. In this context, the overall problem of tracking can be summarized as follows:

1. Detecting every target within a given scene.

2. Distinguishing and categorizing objects as obstacles or objects of interest.

3. Find the centre of gravity of targets within the scene using a hybrid subtractive-K-means clustering technique.

4. Tracking the centre of gravity of the targets within the scene using Extended Kohonen Maps.

5. Tracking the exit targets by the best suitable trackers using a cueing/handoff market-based approach.

5.5 Problem Formulation

With minor variations, the same notation and terminology as those introduced in [49] is used in this thesis. Let:

1. AOI: a two dimensional, bounded obstacle free area of interest.

2. O: a set of n moving targets, $o_j, j = 1, 2, ...n$. The position of o_j at time tt shall be denoted by $o_j(tt)$. Targets cannot leave the area of interest, i.e. $\forall tt \; o_j(tt) \in AOI$.

3. S: a team of m mobile sensors. It is assumed that each sensor carries sensors (laser range finders) to be able to detect the targets.

4. $SC(s_i, tt)$: the sensor coverage, which is the subset of AOI observable by mobile sensor s_i at time tt. This region varies as the sensor s_i moves inside AOI, but its shape is supposed not to vary. In this thesis, an omnidirectional sensing shape for each mobile sensor is assumed.

5. $OM(tt)$: $m \times n$ Observation matrix. A sensor s_i is said to be observing a target when the target is within s_i's sensing range:

$$OM(t) = om_{ij}(tt) = \begin{cases} 1 \text{ if sensor } i \text{ is observing target } j \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{5.1}$$

The goal is to develope an algorithm that maximizes the following metric:

$$AC = \sum_{tt=1}^{t_e} \sum_{j=1}^{n} \frac{g(OM(tt), j)}{t_e} \tag{5.2}$$

where

$$g(OM(tt), j) = \begin{cases} 1 \text{ when there exist } i \text{ such that } om_{ij}(\text{tt}) = 1 \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{5.3}$$

In other words, the problem requires maximizing the number of targets that are observed by the mobile sensors. t_e is the execution time of the algorithm. It is assumed that:

$$\bigcup_{s_i \in S} SC(s_i, tt) \ << \ AOI \qquad \text{for any tt} \tag{5.4}$$

Moreover, it is assumed that the maximum speed of the targets is smaller than the maximum speed of the sensors like assumed in [49, 123] in order to give the sensors the ability to track the targets. If the targets could always move faster, then they could always evade the sensors and the problem becomes insoluble for the mobile sensor team. Finally, it is assumed that the sensors have a communication mechanism, which allows them to send and receive messages in a broadcast mode.

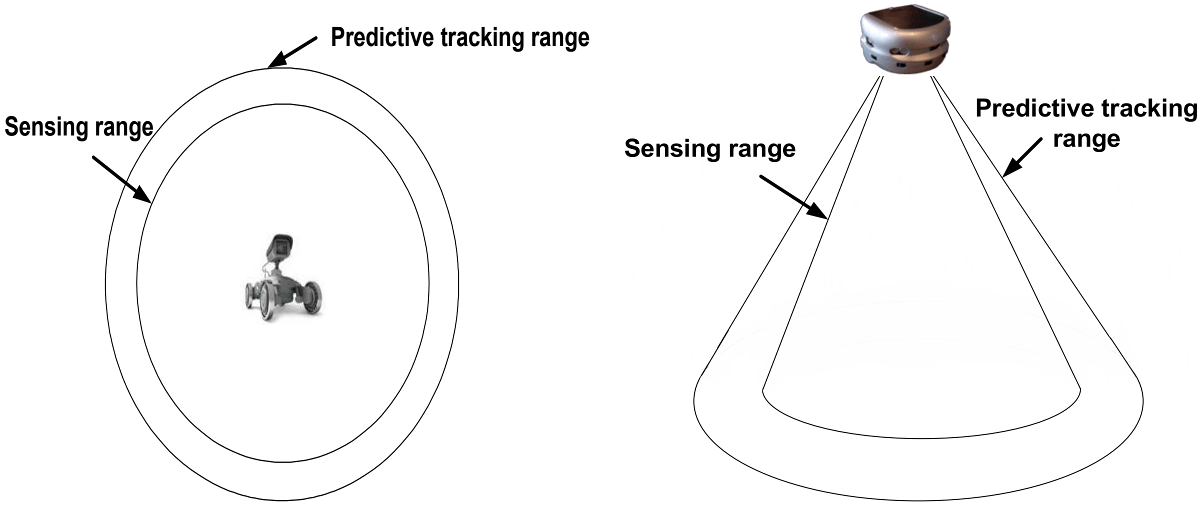
5.6 Approach

As in [49, 123], the sensors are assumed to have an omnidirectional field of view. The proposed approach depends completely on that assumption. The conic field of view will be considered in future research. In the following subsections, more details of the proposed approach are provided.

5.6.1 Target Detection

Ideally, sensor team members would be able to passively detect nearby sensors and targets to ascertain their current positions. Many machine vision algorithms have been developed for this type of position calculation. Although vision systems have a very good resolution, they do not provide accurate distance information [145]. Also, vision-based systems frequently suffer from occlusion and sudden changes in illumination [146]. On the other hand, other types of sensors, such as laser range sensors [145–147] are able to obtain accurate distance and geometric information about the objects in their sensing range. Moreover, laser range sensors do not have the problems of occlusion, and sudden changes in illumination, as with vision systems. The laser range sensors have attracted more attention in the field of detection and tracking [147, 148]. Thus, in this thesis, we propose a detection algorithm that uses laser scanners to detect moving targets and objects. The outputs of the proposed algorithm are the positions of the targets and objects. Each sensor communicates the positions of targets and other objects within its sensing range to its sensor team members.

Fig. 5.1 depicts two types of ranges [123]: the sensing range, and the predictive tracking range. The **sensing range** is the range of the sensor within which it executes the tracking algorithm to maintain observation of the targets. The **predictive tracking range** is the range within which the targets are out of the sensing range but could still be observed (i.e.,



(a)

(b)

Figure 5.1: Field of view (a)Omnidirectional (b) Conic

the target starts to leave the sensor's sensing range).

5.6.2 Find the Center of Gravity

Clustering is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. The following techniques are the most popular clustering techniques, which are used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics:

1. K-means (or Hard C-means) Clustering,
2. Fuzzy C-means Clustering,
3. Mountain Clustering, and

4. Subtractive Clustering.

Among the above-mentioned techniques, subtractive clustering is chosen to initially find the number of clusters, and then these initial clusters are used as initial conditions in K-means clustering to find the centre of gravity of targets within the sensor sensing range. The reason behind using K-means clustering is its high level of accuracy [149].

5.6.2.1 ***Subtractive Clustering***

The subtractive clustering method [150] assumes each data point is a potential cluster centre and calculates a measure of the likelihood that each data point would define the cluster centre, based on the density of surrounding data points. By using it, the quantity of calculation is in proportion to the number of data points and is foreign to the dimensions of problem. However, while the actual cluster centres are not necessarily located at one of the data points, in most cases it is a good approximation, especially with the reduced computation this approach requires.

Consider $m3$ dimensions $n3$ data point $(xx_1, xx_2, ...xx_{n3})$. More generally, we assume that these data points have fallen into a unit hyper box. Because each data point is potential cluster centre, the density of a data point at xx_i is defined as:

$$D = \sum_{j=1}^{n} \exp \frac{\|xx_i - xx_j\|^2}{(r_a/2)^2} \tag{5.5}$$

where r_a is a positive number. Obviously, the data point has the highest potential if the data point xx_i is surrounded by more data points. A radius defines a neighbour area. The data points which exceed raduis r_a, have no influence on the density of data points. The first cluster centre xx_{c_1} is chosen as the point having the largest density value D_{c_1}. Next, the density measure of each data point xx_i is revised as follows:

$$D = D_i - D_{c_1} \exp \frac{\|xx_i - xx_{ci}\|^2}{(r_b/2)^2} \tag{5.6}$$

where r_b is a positive constant, which defines a neighbourhood that has measurable reductions in density measure. Therefore the data points near the first cluster centre xx_{c_1} will have a significantly reduced density measure. After revising the density function, the next cluster centre is selected as the point having the greatest density value. This process continues until a sufficient number of clusters is attained at which all points lie within a loop belonging to a cluster centre.

In this thesis, subtractive clustering will be used to find the initial number and the initial locations of the targets clusters. The parameters r_a, and r_b are selected based on the sensors' field of view specifications in such way to ensure that reaching the centre of gravity of the detected targets will be sufficient to track the clusterd targets. This information will be fed into the K-means clustering technique to find the final locations of target clusters. The reason behind using the K-means clustering technique is its high accuracy compared to subtractive clustering.

Each mobile sensor tries to maximize the number of targets inside its sensing range. This is done by positioning itself at the center of gravity **CG** of the targets inside its sensing range. In order to approach this position, an **Extended Kohonen Neural network** will be used. As the targets move inside the sensor sensing range, the CG changes and so the mobile sensor will move reactively.

5.6.2.2 ***K-means Clustering***

K-means [151] is one of the simplest unsupervised learning algorithms that solves the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed *a priori*. The main idea is to define k centroids, one for each cluster. These centroids should be placed in intelligently, because different location will give different results. So, the best choice is to place them as far as possible from one another. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as varying centres of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words, centroids don't move any more. Finally, this algorithm aims at minimizing an *objective function*, in this case, a squared error function. The objective function:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| xx_i^j - c_j \right\|^2 \tag{5.7}$$

where $\left\| xx_i^j - c_j \right\|^2$ is a chosen distance measure between a data point xx_i^j and the cluster centre c_j, which is an indicator of the distance of the n data points from their respective cluster centres. The partitioned groups are defined by a $c \times n_4$ binary membership matrix MM, where the element mm_{ij} is 1 if the j data point xx_j belongs to group i, and 0 otherwise. Once the cluster centres c_i are fixed, minimizing mm_{ij} for Eq. 5.7 can be derived as follows:

$$mm_{ij} = \begin{cases} 1 \text{ if } \ \left\| xx_j - c_i \right\|^2 \leq \left\| xx_j - c_k \right\|^2 \ \text{ for } k \neq i \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{5.8}$$

Which means that xx_j belongs to group i if c_i is the closest centre among all centres. On the other hand, if the membership matrix is fixed, i.e., if mm_{ij} is fixed, then the optimal centre c_i that minimizes Eq. 5.8 is the mean of all vectors in group i:

$$c_i = \frac{1}{GR_i} \sum_{k, xx_k \in GR_i} xx_k \tag{5.9}$$

The algorithm is presented with a data set xx_i, i $= 1, \ldots\ldots$n; it then determines the cluster centres c_i and the membership matrix MM iteratively using the following steps:

- **Step 1:** Initialize the cluster centre, c_i, i $= 1, \ldots\ldots$c. This is typically done by randomly selecting c points from among all of the data points.
- **Step 2:** Determine the membership matrix MM.
- **Step 3:** Compute the cost function. Stop if it is either below a certain tolerance value or if its improvement over the previous iteration is below a certain threshold.
- **Step 4:** Update the cluster centres. Go to step 2.

This is a simple version of the K-means procedure. It can be viewed as a greedy algorithm for partitioning the n_4 samples into k clusters so as to minimize the sum of the squared distances to the cluster centres. The results depend on the value of k, which must be specified in advance (from subtractive clustering).

5.6.3 Extended Kohonen Neural Network

Kohonen's self-organizing map (SOM) network [152] is an unsupervised learning neural network that maps an n-dimensional input data to a lower dimensional output map while maintaining the original topological relations. The extended SOM network further groups the nodes on the output map into a user-specified number of clusters. As shown in Fig. 5.2, the Kohonen Map is a two-layered network consisting of an input layer of neurons directly and fully connected to an output layer. Typically, the output layer is organized as a two-dimensional grid; w_r is the fan-in weight vector (reference vector) associated with the neuron placed at position r on the grid.

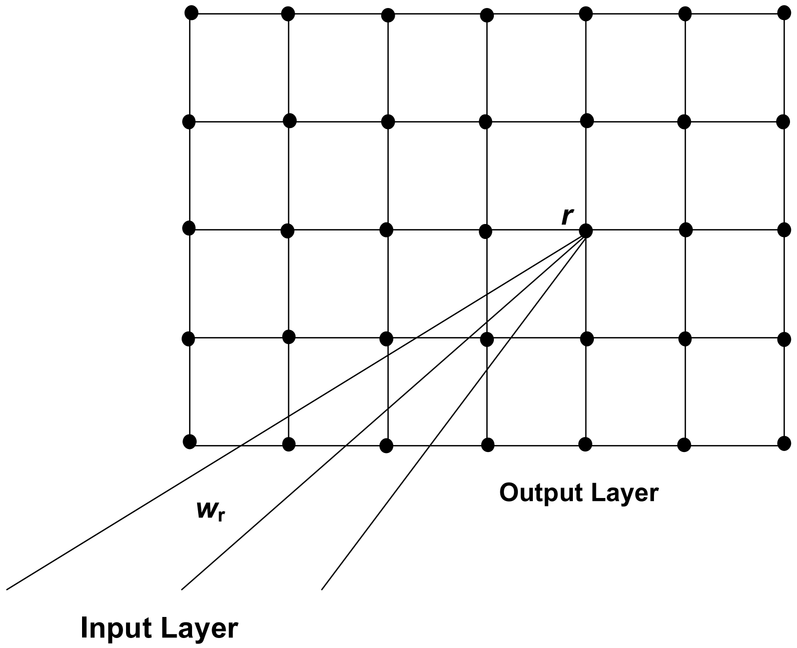


Figure 5.2: Kohonen Map

The network is trained by unsupervised learning on a set of examples. For each example x, the following sequence of steps is executed:

1. x is presented to the input layer.
2. A competition between the neurons takes place. Each neuron calculates the distance between its reference vector w_r and input pattern x.

$$d(x, w_r) = \|x - w_r\|^2 \tag{5.10}$$

The neuron ne whose weight vector is the closest to x is the winner of the competition.

$$ne = \arg\ \min d(x, w_r) \tag{5.11}$$

3. ne is awarded the right to learn the input pattern, i.e. to move closer to it in data space:

$$w_{ne}^{new} = w_{ne}^{old} + \alpha(x - w_{ne}^{old}) \tag{5.12}$$

where w_{ne}^{old} is the fan-in weight of the winner ne before learning, w_{ne}^{new} is the fan-in weights of the winner ne after learning, and α is the learning rate.

A special trait of the Kohonen algorithm is that the learning step is extended also to the neighbors of the winner ne. The neighbors of ne are those output elements whose distance to ne, measured on the grid G, is not greater than a neighborhood parameter $n(t)$. The Kohonen Map can be ***extended*** by adding a fan-out weight vector z_r to store the neuron output value as shown in Fig. 5.3. Adding a fan-out to the output neurons gives the Extended Kohonen Map the ability to learn by supervised learning.

The computation in the EKM network proceeds as follows: When an input pattern x is presented to the input layer, the neurons on G compete to respond to it. The competition involves the neurons' fan-in weight vectors w_r, and consists of the computation of the distance between x and each w_r. The neuron ne, whose fan-in vector w_{ne} is the closest to x, is the winner of the competition, and its fan-out vector z_{ne} is taken as the network output answer to x. During the training phase, both the input pattern x and the desired output value y proposed by the teacher are learnt by the winning neuron and by its neighbors on

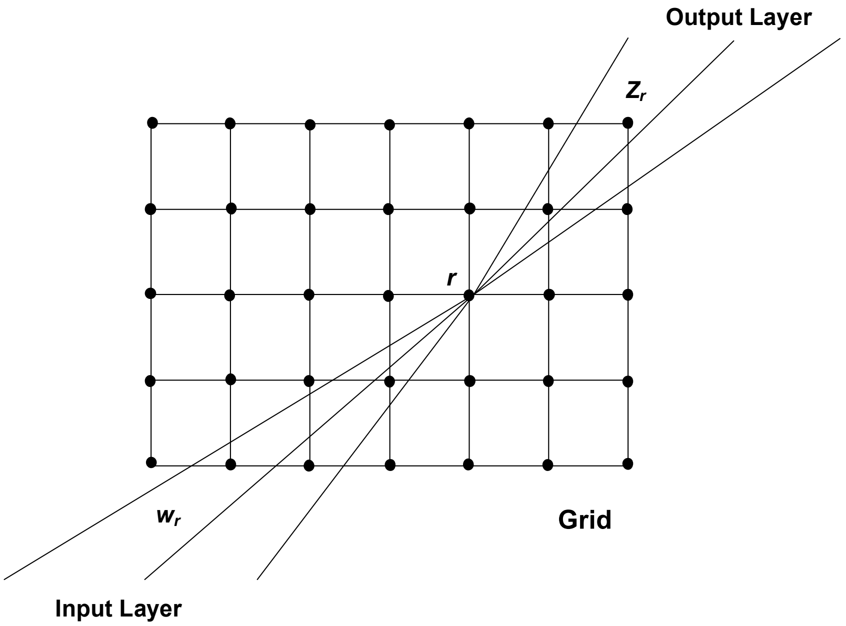


Figure 5.3: Extended Kohonen Map

the grid. The learning step consists of moving the fan-in weight vectors of the selected neurons closer to x, and their fan-out weight vectors closer to y.

5.6.3.1 Supervised Extended Kohonen Neural Network

The term 'supervised' originates from the fact that the desired signals on individual output nodes are provided by an external 'teacher'. This is usually performed with feedforward nets where training patterns are composed of two parts, an input vector and an output vector, associated with the input and output nodes respectively. A training cycle consists of the following steps: an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done and the errors, between the desired and actual response for each node in the output layer, are found. These are then used to determine weight changes in the net according to the prevailing learning rule.

In the supervised Extented Kohonen Map, the situation is a little bit different from that of feedforward nets mentioned above. During the training phase, both the input pattern x and the desired output y proposed by a teacher are learned by the winning neuron and by its neighbors on the grid. The learning step consists of moving the fan-in weight vectors of the winner (which is the closet neuron to the input pattern x) closer to x, and their fan-out weight vectors closer to y. This learning style has been described as a *competitive-cooperative* training rule [153]. It is competitive because the neurons compete through their fan-in weight vectors to respond to the presented input pattern. As a consequence, only that part of the network which is relevant to deal with the current input data undergoes the learning process. The rule is also cooperative in that the output value learned by the winning neuron is partially associated to the fan-out weight vectors of its neighbors. The supervised Extended Kohonen Map learning is shown in Fig. 5.4.

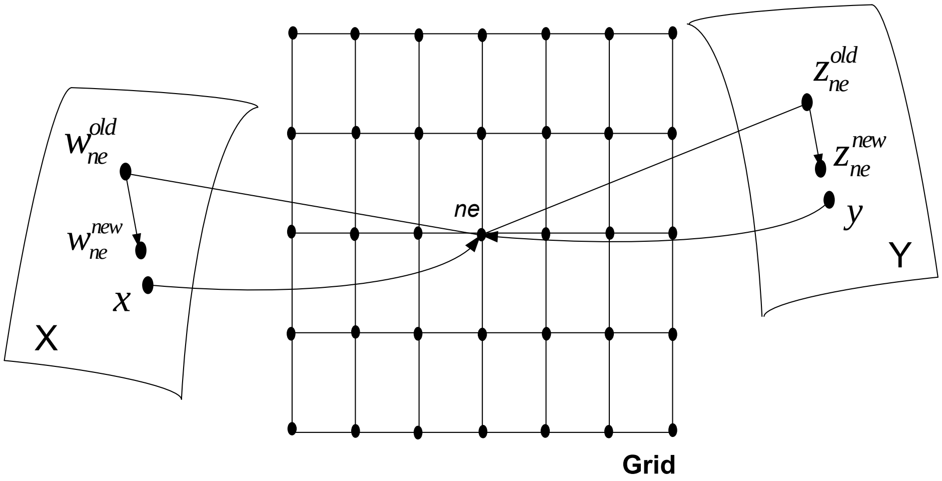


Figure 5.4: supervised Extended Kohonen Map learning

As shown, suppose we have trained an EKM on a set of examples $< x; y >$, where x is a point in the input data space X and y is a point in the output data space Y. The input pattern x is presented to the network input layer to retrieve the corresponding output value, which will be an approximation to y. This is accomplished in the usual way, namely by letting the fan-in weight vectors w_r compete on x, and by taking the fan-out weight vector z_{ne} of the winner ne (which has the closet fan-in weights to x) as the network output value. As mentioned above, ne is awarded the right to learn the input pattern x by moving closer to it. Also, the fan-out z_{ne} is awarded the right to learn the output pattern y by moving closer to it:

$$z_{ne}^{new} = z_{ne}^{old} + \alpha(y - z_{ne}^{old}) \tag{5.13}$$

where z_{ne}^{old} is the fan-out weights of the winner ne before learning, z_{ne}^{new} is the fan-out weights of the winner ne after learning, and α is the learning rate.

5.6.3.2 Unsupervised Extended Kohonen Neural Network

The term ‘unsupervised’ originates from the fact that no output data is provided (i.e., there is no supervisor). In unsupervised learning the machine simply receives inputs $x_1, x_2, ...$ but obtains neither supervised target outputs as in supervised learning, nor rewards from its environment as in reinforcement learning. The only thing that unsupervised learning methods have to work with is the observed input pattern x_i, which is often assumed to be independent samples. It may seem somehow mysterious to imagine what the machine could possibly learn given that it does not get any feedback from its environment. However, it is possible to develop a formal framework for unsupervised learning based on the notion that the machine’s goal is to build representations of the input that can be used for decision making, and for predicting future inputs.

The best-known form of unsupervised learning in the neural network literature is Kohonen’s self-organizing map (SOM) [152]. In this thesis, the Extended Kohonen Map as unsupervised learning network [115] is used to get the function of mapping from the space of sensory data to the space of motor commands for tracking purposes. The details of the single target tracking process is shown in algorithm 1.

5.6.4 The Proposed Cooperative Multi-target Tracking Approach

In multisensor tracking systems, tracking performance can be improved by maximizing cooperation between sensors in response to the changes in target motion. As mentioned in section 5.2.4, there are two kinds of cooperation: uncertainty reduction and task allocation. The second type of cooperation is considered in this thesis. The mobile sensors used in this work are assumed to be equipped with high-bandwidth communications and an array of sensors and actuators, which give the sensors the ability to achieve cooperative behavior at the group level. A cueing/handoff market-based method like the one in [16] is used to guarantee that there is only one mobile sensor that will respond to the help call coming from the sensor that detects a target about to exit its sensing range (or in its predictive tracking range). The mobile sensor that detects an exiting target will do the task of an auctioneer (i.e., every tracker can do the function of the auctioneer) This makes our algorithm more robust than methods that use only one coordinator because there is no central point of failure in this case. Using a cueing/handoff market-based algorithm will guarantee that the most suitable sensor will track the exit target. In other words, there is no need to explore the environment or to check the answered help calls as in [49]. This makes the proposed method a more energy-efficient one than the method in [49]. The cueing/handoff market-based method proceeds as follows:

1. **Help cueing**: the mobile sensor that needs help (detects an exiting target) broadcasts a help call to its teammate. One sensor can issue multiple help calls according to the number of targets about to be lost. Each help call includes the position of the target to be lost.

Algorithm 1: Single Target Tracking Algorithm

```
   Data: $\theta$: the angle difference between the detected target and the mobile tracker $s$;
         $dis$:the distance between the detected target and the mobile tracker $s$;
         $Nr$: number of mobile sensors;
   Result: $(speed, angle)$ the speed and orientation angle of the tracker;
1  begin
2      Initialize the parameters (fan-in and fan-out weights) of the Extended Kohonen
       Map from supervised learning mentioned above;
3      repeat
4          Determine the neuron $ne$ whose input weights $(\theta_{ne}, dis_{ne})$ are the nearest
           weights to the detected target sensory input $(\theta, dis)$;
5          Determine the output activity $(speed(ne), angle(ne))$ of the winning neuron
           and move the tracker;
6          Compute the actual displacement of the tracker $v = (\theta_s, dis_s)$, which is the
           difference between the old sensory input and the new one;
7          Use $v$ as a training input to determine the winning neuron (step 2 but with $v$
           as an input);
8          Adjust the input weights $w_i$ in the neighborhood of the winning neuron $ne$:

                   $\Delta w_i = \alpha Gaus(ne, i)(v - w_i)$                    (5.14)

           Where $Guas(ne, i)$ is the Gaussian function of the distance between the
           winning neuron and input neurons and $\alpha$ is the learning rate;
9          Adjust the output weights in the neighborhood of the winning neuron to
           minimize the error between the actual activity $(speed(s), angle(s))$ of the
           mobile tracker and the output of the Extended Kohonen Map
10     until *self organization is reached (at which the error is minimum)*;
```

2. **Bid submission**: after each sensor receives the help call, they send their bids to the auctioneer (the sensor that issued the help call). The bids include the distances of the bidders to the target.

3. **Close of auction**: the auctioneer processes the bids, determines the winner, and notifies the bidders with a message which indicates who is the winner.

4. **Task handoff**: the winner will proceed by orienting itself to face the target and move with its maximum speed to put the target inside its sensing range.

The existence of the auctioneer does not mean that the proposed system is completely centralized. The system is still distributed and the function of the auctioneer is only to start the action of tracking. Once started, the tracker will do its task on its own, without any dependency on the auctioneer. Furthermore, there is no single point of failure because the auctioneer changes in the system according to the sensor that requests help (i.e., the sensor that requests help will be the auctioneer). The process of the cueing/handoff market-based algorithm is shown in Fig. 5.5.

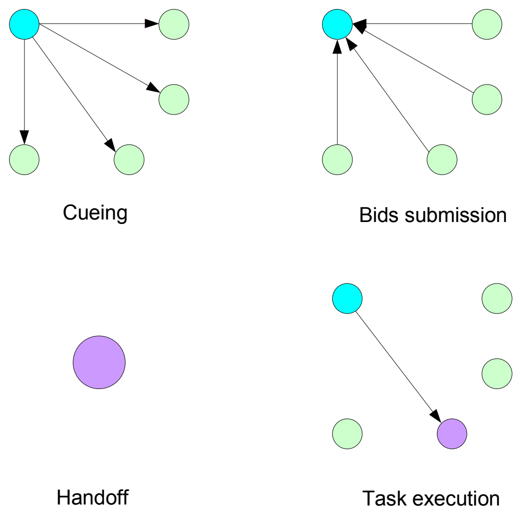


Figure 5.5: Cueing/Handoff market-based algorithm

The cooperative tracking capability of a team of mobile sensors, each fitted with Extended Kohonen Maps, is utilized to maximize the coverage of multiple mobile targets. The main goal is to develop an energy-efficient cooperative tracking strategy. To achieve this goal, a minimum number of mobile sensors is used to track the moving targets in the environment. The whole tracking strategy may be summarized in the following steps:

1. Clustering the targets in the environment by using a hybrid subtractive–K-means clustering technique as mentioned above.
2. The clusters' locations are reached by the best suitable trackers (the closest mobile sensors to the clusters' centres). The trackers use Extended Kohonen Maps to reach these locations.
3. Each mobile tracker will try to maximize the number of targets inside its sensing range by approaching the centre of gravity **CG** of the moving targets.
4. Approaching this location will be achieved through Extended Kohonen Maps as mentioned above.
5. Each mobile sensor will have three modes of operation:
    (a) **Follow**: in this mode, the mobile tracker will follow the **CG** of the targets in its sensing range.
    (b) **Help**: when any of the sensors observe any of the targets leaving the sensing range and entering the predictive tracking range, it predicts that this target will be lost, and sends a help call asking for help from other sensors.
    (c) **Sleep**: when no help call is requested and no targets are detected, the mobile sensor is asleep. In other words, none of the sensors is doing the function of exploring the environment.

6. When a mobile sensor detects a target entering its predictive tracking range, it sends a help call to its teammates to track the exit target. The cueing/handoff market-based algorithm mentioned above is used to find out the best suitable sensor to track this target. The best suitable sensor is selected only from the sensors that are in sleep mode (i.e., no targets to track). If no trackers are in sleep mode, this target will be lost. The existence of the exit target in the predictive tracking range of the tracker gives a chance for this target to be observed for a period of time before being tracked by the tracker selected by the auctioneer. This means that the time of inobservance is minimized.

7. The sensors that are observing other targets are out of the game (i.e. they don't send any bids) unless they realize that they can observe the new target as well as targets they are currently observing.

8. If at any time before being tracked by the auction winner, the exit target enters the sensing range of another busy tracker, this tracker will send a message indicating that the target is now tracked and there is no need for more tracking. This message will stop the tracker selected by the auctioneer from reaching this target.

9. Each mobile sensor communicates with its teammates the positions of the targets it observes. So, if any of the sensors sees that it can observe new targets besides the target or targets it currently observes (i.e., the new and the old targets are clustered in one cluster that can be observed by this sensor), it sends a bid for this exit target.

5.7 Conclusion

In this chapter, a cooperative multi-target tracking strategy has been proposed for surveillance and security applications using a team of mobile sensors. The Extended Kohonen

Map is utilized for single target tracking and multi-target tracking. The presented target tracking methodology has decoupled the single target tracking algorithm from the cooperation strategy. The hybrid subtractive–K-means clustering technique has been used to cluster the targets in the given environment to minimize the number of active sensors and thus save energy. The cueing/handoff market-based algorithm has been utilized to coordinate the performance of the trackers while tracking the moving targets. Using the cueing/handoff market-based method has enhanced energy efficiency of the proposed multi-target tracking strategy to be an energy-efficient one.

Chapter 6

Experimental Results

As mentioned previously in this thesis, three phases should be considered in order to accomplish the surveillance mission: task allocation, target detection, and target tracking. In this chapter, the simulations and results of these phases are presented in detail. The task allocation simulations and results are presented first in section 6.1. This is followed by the target detection and tracking simulations and results in section 6.2. Finally, some conclusions are drawn in section 6.3.

6.1 Complex Task Allocation Simulations and Results

In order to evaluate the proposed task allocation approach, an area surveillance application is considered, where the goal is to monitor some areas in a public place using a team of mobile sensors, each equipped with a vision system and a laser range sensor. To tackle this application, it is assumed that for each area, a set of surveillance points (vantage points) is selected from which the mobile sensors can view the interior of the area as illustrated in Fig. 4.2. The architecture under study achieves the surveillance task while keeping in mind the minimization of the total traveling distance of the whole team. Sensors frequently use the SSP algorithm when bidding, and when reordering schedules after trades or task completion.

6.1.1 Outdoor Scenario

The initial experiments are performed within a 2D simulation environment using Player/Stage simulator [154, 155]. An outdoor scenario is considered first to assess the proposed task allocation approach. An example of this scenario is a battlefield in which it is required to know where the enemy is. In order to address this, the mobile sensors team divides the battlefield into various areas. A set of surveillance points is selected from which the mobile sensors can view the interior of each area. To simplify this scenario , an example of an outdoor scenario was run with six different cases as shown in table 6.1 with 50 runs for each case. These runs consider different-sized areas, different locations, and different intial positions for the mobile sensor team. In this type of scenarios, the mobile sensors are permitted to move from one vantage point to another passing into the areas of interest. The results of this scenario, considering all previously mentioned algorithms in chapter 4, are shown in table 6.2. The average cost is used as an evaluation metric for this comaprison. The average cost is computed by calculating the cost of executing the mission task using 50 runs and then taking the average.

As seen in table 6.2, for both allocation types (centralized & hierarchical), using two plans (2p) is better than using one plan (1p) for both search trees.

To assess the proposed task allocation algorithms, the proposed fixed and dynamic tree allocation algorithms are compared with a higher level allocate-then-decompose, allocate-then-decompose, and decompose-then-allocate algorithms mentioned in section 4.3.3. In terms of this average cost, the results in Fig. 6.1, and Fig. 6.2 show that both fixed and dynamic tree allocations (centralized and hierarchical) consistently outperform the other algorithms, especially for the complex cases such as cases 4, 5, and 6. It can also be seen

Table 6.1: Test Cases

Case	Number of areas	Number of mobile sensors
1	1	2
2	4	2
3	5	3
4	7	5
5	6	6
6	4	7

that the dynamic tree allocation outperforms fixed tree allocation, which was expected as the replanning ability is added to the sensors in the dynamic tree allocation.

On average, the hierarchical task tree algorithm is better than the centralized task tree algorithm, besides its good feature of relying on different auctioneers, compared to one auctioneer in the centralized algorithm. This is because the hierarchical auctioning allows more auction rounds to take place, and so the mobile sensors may find themselves in better positions to win more tasks than their old positions if it is found beneficial to the whole system. Thus the hierarchical auctioning increases the possibility of improving the system performance than the centralized auctioning does.

In order to determine how the proposed tree allocation algorithms are affected by problem complexity, two sets of experiments were conducted. In the first set, the number of mobile sensors was varied while the number of areas was kept fixed, while in the second set, the number of areas is varying while the number of mobile sensors is kept fixed. The results for both sets (shown in Fig. 6.3, and Fig. 6.4) are deduced using the average of 50 trials. The quality of the proposed algorithms is again evaluated using the average traveling cost as can be seen in the experiments shown in Fig. 6.1, and Fig. 6.2. For each type of allocation

Table 6.2: Average cost for outdoor task allocation

	Centralized Allocation				**Hierarchical Allocation**			
	Fixed Tree		**Dynamic Tree**		**Fixed Tree**		**Dynamic Tree**	
	BF(2p)	**BF(1p)**	**BF(2p)**	**BF(1p)**	**BF(2p)**	**BF(1p)**	**BF(2p)**	**BF(1p)**
Case 1	5.41503	5.47825	5.31439	5.33848	5.41503	5.47825	**5.31439**	5.33848
Case 2	10.0589	11.4637	9.35118	10.69195	9.69662	11.0861	**9.02721**	10.20856
Case 3	11.0037	12.355	10.3924	11.413	10.8073	11.8633	**10.213**	11.2352
Case 4	16.9887	18.1871	15.7433	17.1211	16.0556	17.5086	**14.3717**	16.0749
Case 5	18.0205	19.4568	15.9674	17.5847	17.3766	18.5274	**15.0506**	16.6805
Case 6	20.4885	21.6675	18.9887	20.195	18.6251	19.954	**17.0559**	18.3254

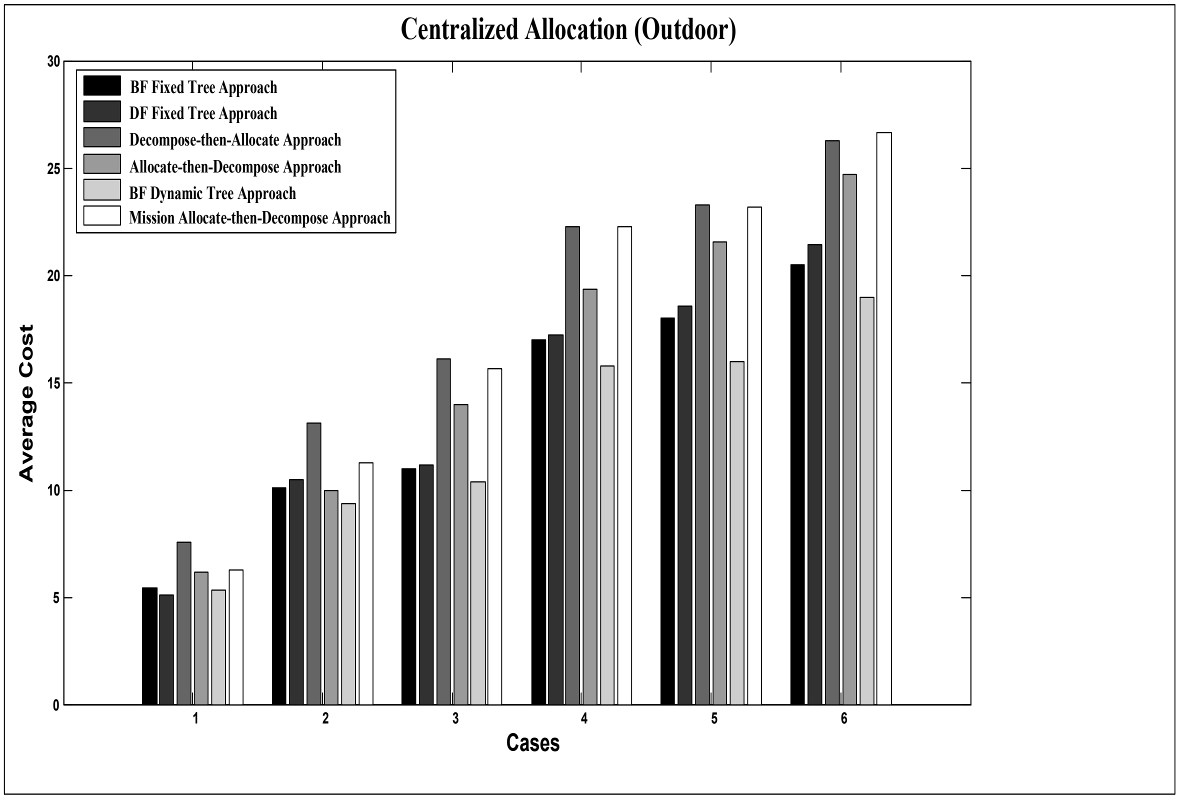


Figure 6.1: Comparison of the average cost for centralized allocation mechanism for outdoor environment

mechanism (centralized and hierarchical), as the number of mobile senosrs increases in the first set shown in Fig. 6.3, the average traveling cost decreases. It can also be seen from Fig. 6.3 that the average traveling cost is increased as more areas are incorporated into the environment. For the second set of experiments shown in Fig. 6.4, as the number of areas increases, the average traveling cost increases. Also, the average traveling cost is decreased as more mobile sensors are used to achieve the surveillance mission. A general trend in the results shown in Fig. 6.3, and Fig. 6.4 is that the hierarchical tree allocation outperforms the centralized one.

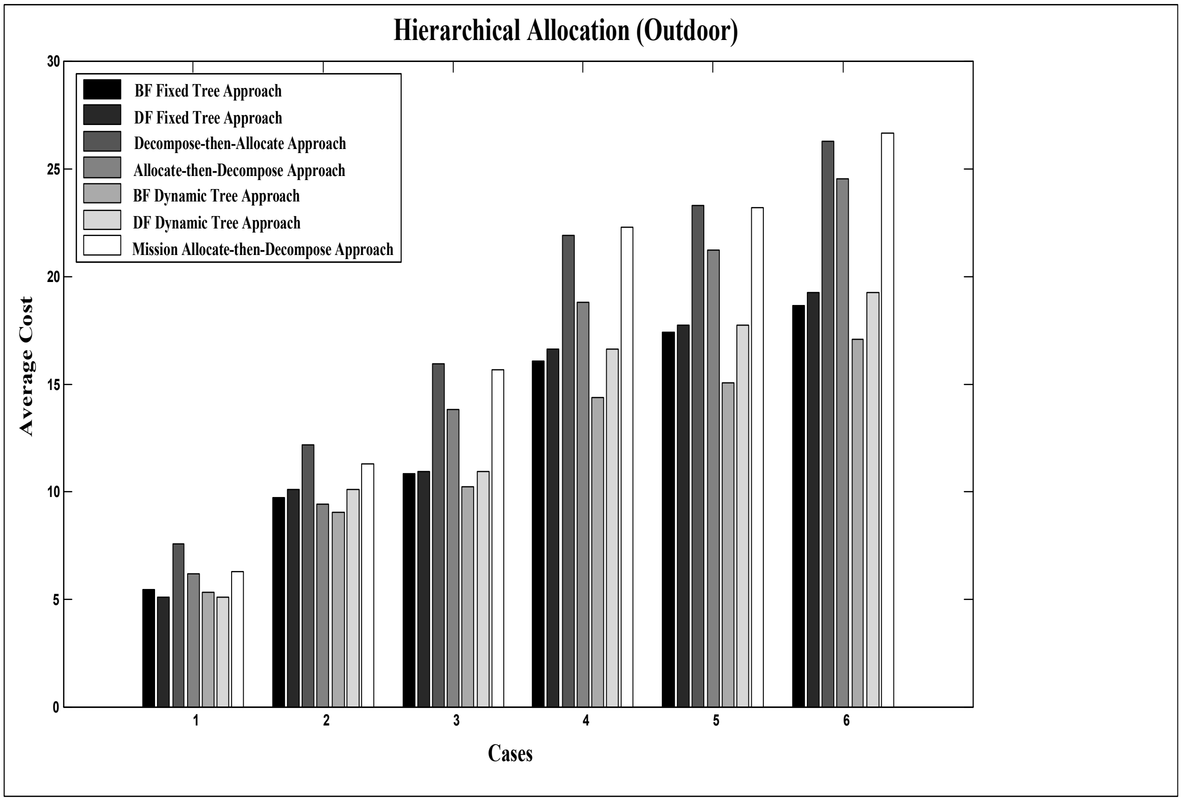


Figure 6.2: Comparison of the average cost for hierarchical allocation mechanism for outdoor environment

6.1.2 Indoor Scenario

This subsection describes a surveillance simulations and results for an indoor scenario. An example of this is the use of a team of mobile sensors to survey an indoor environment, such as malls or airports. Fig. 6.13 shows a simplified plan view of Waterloo airport in the city of Waterloo, Ontario, Canada. As shown, the Waterloo airport consists of six main areas, and so the goal of the proposed system is to track targets within these areas, such as people, in order to secure the airport. In order to accomplish this, the airport areas (areas of interest (AOIs)) should first be allocated to the available mobile sensors. Each mobile sensor will scan the allocated area, if any, looking for targets to track. Not all sensors will

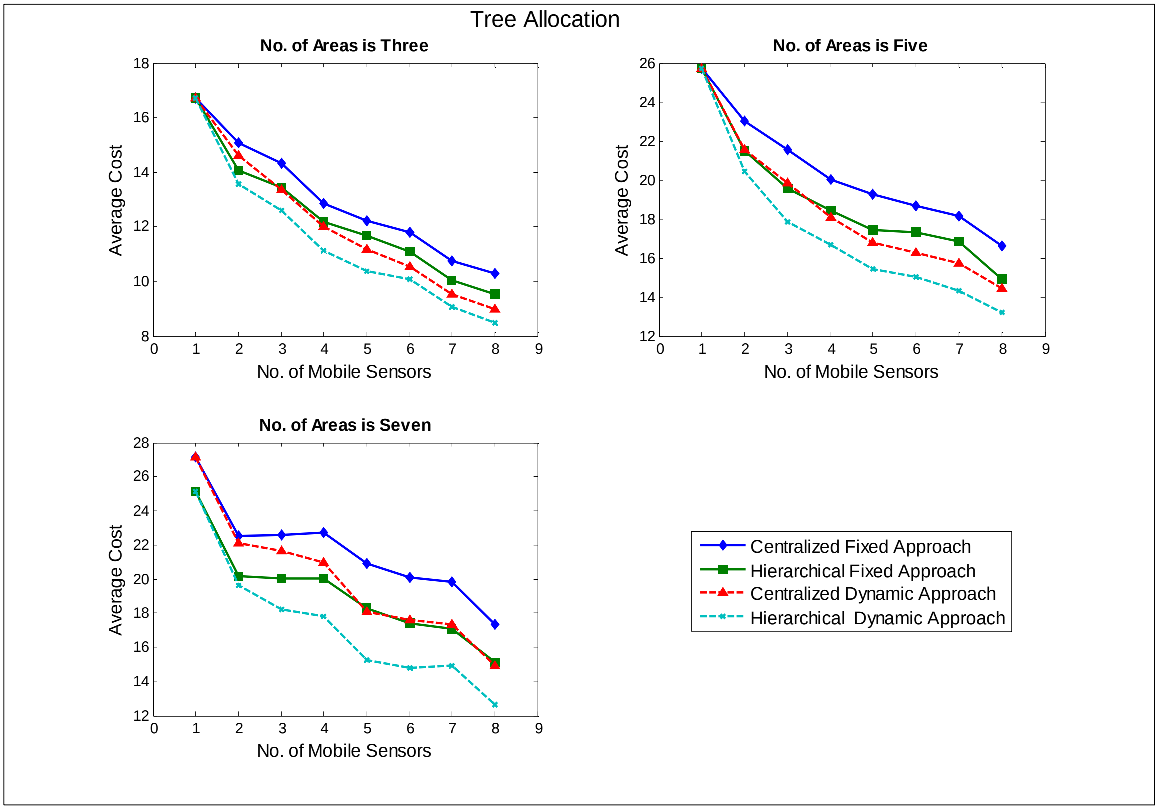


Figure 6.3: A comparison of the solution quality of task tree allocation varying the number of mobile sensors

have areas to scan. In other words, the proposed task allocation approach may allocate more than one area to one mobile sensor in order to minimize the traveling cost. The number of areas of interest in this scenario is fixed (six areas), which is different than the outdoor scenario mentioned above. So, the presented simulations in this section assumes a fixed number of areas with varying the number of mobile sensors required to achieve the surveillance mission.

The results of this scenario, considering all task allocation algorithms mentioned in chapter 4, are shown in table 6.3. The experiments are performed within a 2D simulation environment using Player/Stage simulator. As seen in table 6.3, different numbers of

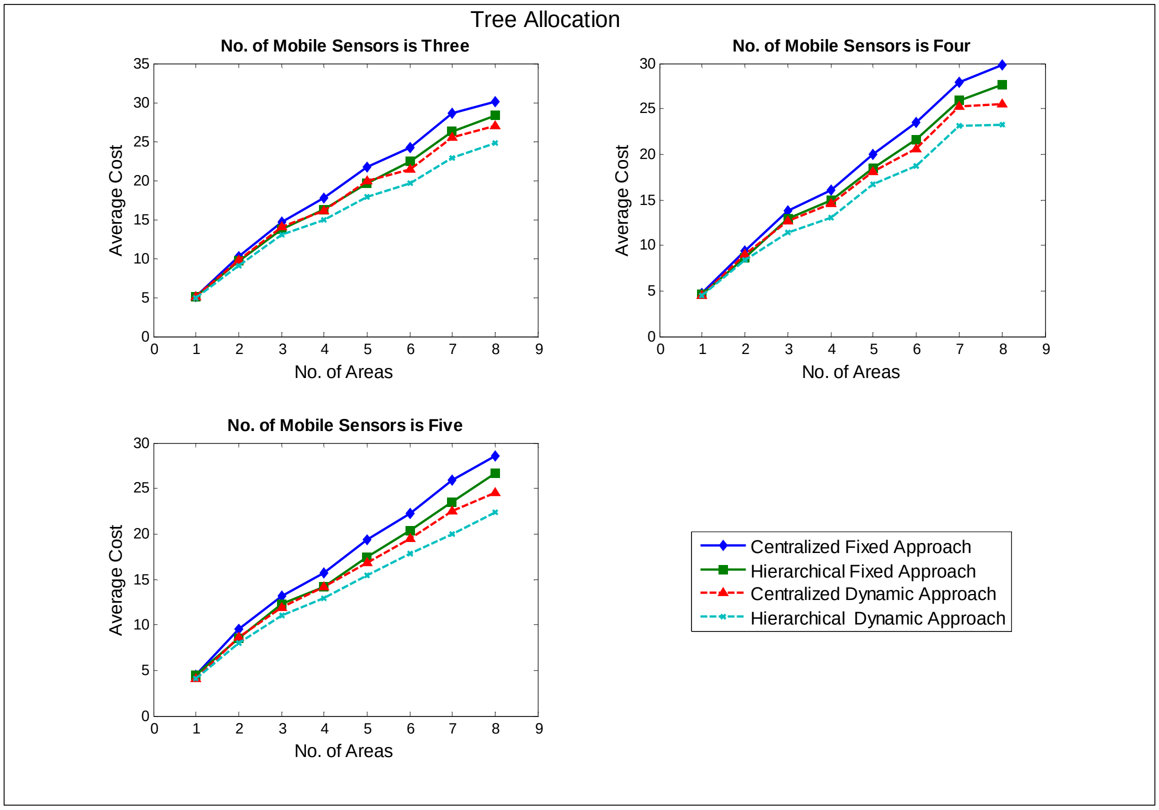


Figure 6.4: A comparison of the solution quality of task tree allocation varying the number of areas

mobile sensors are considered: starting from two sensors and ending with eight sensors. Each datum in table 6.3 is gotten by calculating the average of 50 runs. These runs consider different initial mobile sensors positions. A general trend in these simulations is that for both allocation types (centralized & hierarchical), using two plans (2p) is better than using one plan (1p) for both search trees.

The proposed task allocation algorithms are compared with the existing task allocation approaches, as was done in the outdoor scenario. The results in Fig. 6.5, and Fig. 6.6 show that both fixed and dynamic tree allocations (centralized and hierarchical) consistently outperform the other algorithms. It can also be seen that the dynamic tree allocation

Table 6.3: Average cost for indoor task allocation

	Centralized Allocation				**Hierarchical Allocation**			
	Fixed Tree		**Dynamic Tree**		**Fixed Tree**		**Dynamic Tree**	
	BF(2p)	**BF(1p)**	**BF(2p)**	**BF(1p)**	**BF(2p)**	**BF(1p)**	**BF(2p)**	**BF(1p)**
2 Sensors	110.721	111.915	107.754	111.585	108.851	109.989	**103.535**	107.825
3 Sensors	101.33	102.409	98.1727	101.607	98.2938	99.2916	**93.6337**	97.9088
4 Sensors	96.3275	98.5558	90.9356	95.8027	93.054	94.4705	**84.2702**	88.4433
5 Sensors	89.1474	91.32	85.1072	88.3901	84.5989	85.7978	**79.9255**	82.8419
6 Sensors	86.7868	89.3362	82.5283	86.5057	80.6873	85.7978	**74.1832**	77.564
7 Sensors	81.9689	83.7445	78.094	82.2521	75.3493	76.5879	**70.0992**	73.4569
8 Sensors	76.9689	78.7445	73.094	77.2521	70.3493	71.5879	**65.0992**	68.4569

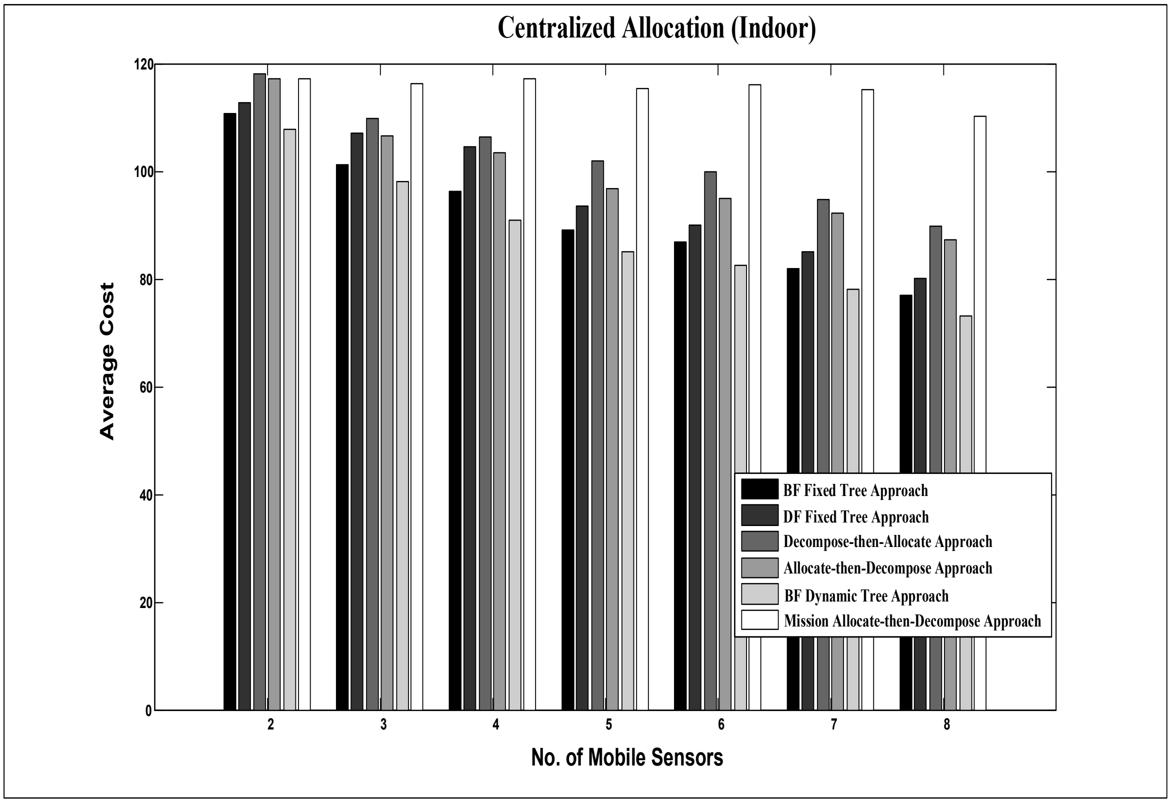


Figure 6.5: Comparison of the average cost for centralized allocation mechanism for indoor environment

outperforms fixed tree allocation. The reason for this is the replanning ability, which is added to the sensors in the dynamic tree allocation.

6.2 Target Tracking Simulation and Results

As mentioned in chapter 5, the target tracking algorithm for an individual target is decoupled from the cooperative tracking algorithm for a multi-tracking system. So, the simulation results for single target tracking system, as a basis layer of the cooperative multi-target system, are presented first. The target tracking algorithm design depends on

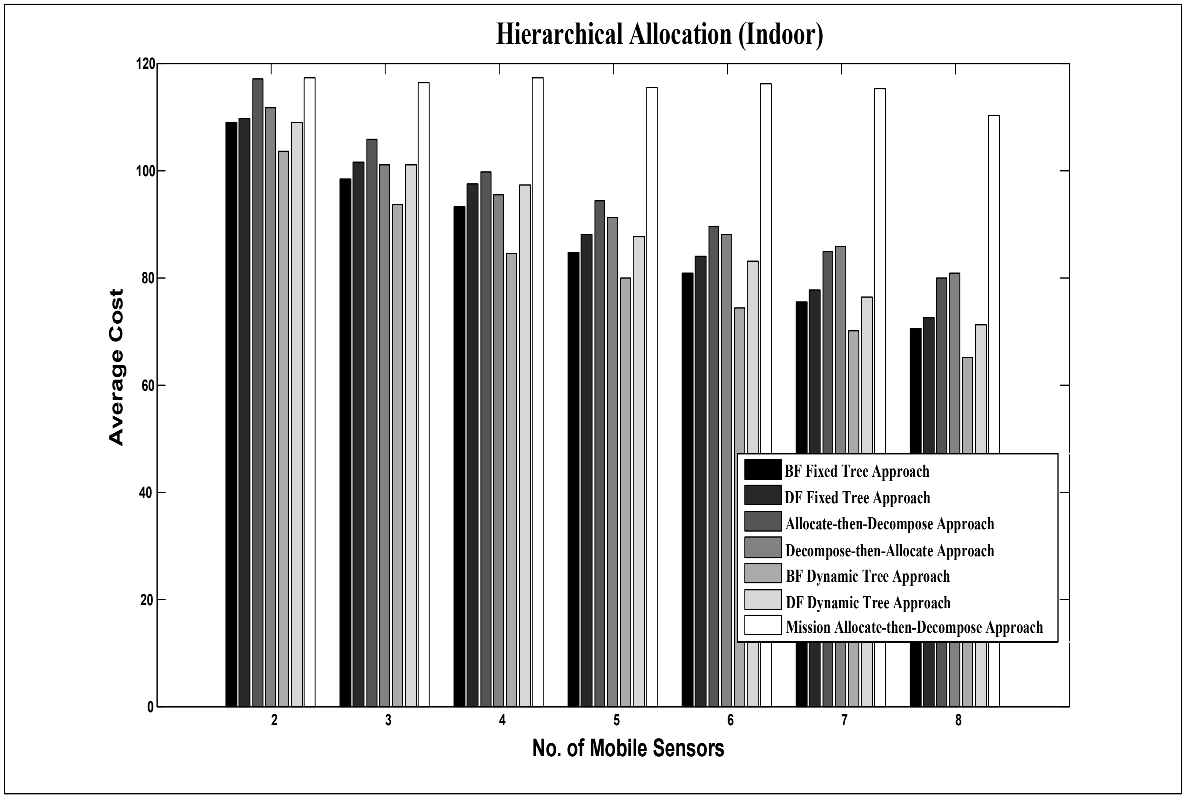


Figure 6.6: Comparison of the average cost for hierarchical allocation mechanism for outdoor environment

targets, sensors, and environments. Target motion recognition is one of the fundamental capabilities that a mobile sensor must have in order to operate in an environment. Therefore a single tracker which tracks and reports the positions of moving targets in the vicinity of this tracker is designed. The simulation results of two types of trackers are presented – supervised and unsupervised Extended Kohonen Maps. The tracking performance is judged by both accuracy of tracking and speed of convergence. Also, the simulation results for a cooperative multi-target tracking system are presented.

6.2.1 Tracking using Supervised Learning EKM

The first step in tracking a target using Extended Kohonen Maps is to estimate the parameters of the map. This can be done by using supervised learning. First, $<$ perception; action $>$ pairs are collected by observing the sensor's behavior during operation. Then, a learning by examples method is used to estimate the parameters of the model, which is the Extended Kohonen Map. To be more specific, an example is a pair $< (s_r, s_l), (\theta, dis) >$. (s_r, s_l) are the angular velocities of the left and right wheel, while (θ, dis) are the sensor's heading change and the distance traveled by its axle mid-point during a fixed time Δt respectively. In [156], the authors observed that this neural network model has the property of being naturally invertible. Given an input pattern, the network output value is retrieved by competition on the neuron fan-in weight vectors. This is the standard use of the input-output mapping, and is called forward mode. Also, given an output value, a corresponding input pattern can be obtained by competition on the neuron fan-out weight vectors. This is called backward mode. The invertibility property makes the Extended Kohonen Map worth considering for sensory motor modeling.

In this section, the EKM is trained on the forward mode, namely on a transformation from the space of motor commands to the space of visual perceptions. Now the trained network is used in backward mode to compute the inverse function, which transforms a visual perception into a motor command. The task is to guide the mobile sensor to a target location placed at arbitrary angle θ and distance dis in the workspace. The angle and distance data are provided by the laser range sensor attached to the tracker. To be more specific, θ is now defined as the angle between the sensor's heading direction and the vector connecting the sensor axle mid-point and the target, while dis is the Euclidean distance between the sensor axle mid-point and the target location. The observed θ and dis values are supplied to the EKM in backward mode to retrieve a velocity pair (s_r, s_l). For this particular application, the competition on the pattern (θ, dis) has been designed to consider its components in sequence. First, θ is processed. Thus, the competition is restricted to the weight vector component of the neuron, which stores the angle information. As the result of this preliminary step, a subset of grid neurons which match θ equally well is selected. Second, dis is processed, but only on neurons selected at the previous step. The competition is restricted to the weight vector component of the neuron which stores the distance information. The overall competition process leads to the selection of a velocity pair for the mobile sensor, namely the fan-in weight vector of the winning neuron. In the simulations introduced next, 30x30 neurons are used in the structure of the Extended Kohonen Map.

Fig. 6.7 shows tracking a moving target using supervised learning EKM. The small circle and square are the initial positions of the target and the tracker respectively. The indicated arrows shows the direction of movement of both, the target and the tracker. As seen from the trajectories of the target and the tracker, the supervised learning EKM does not provide a good tracking performance in terms of accuracy and tracking speed.

6.2.2 Tracking using Unsupervised Learning EKM

In this case, the same structure for the extended Kohonen map is used but without any idea about the correct parameters of it (i.e., the fan-in and fan-out weights). Random

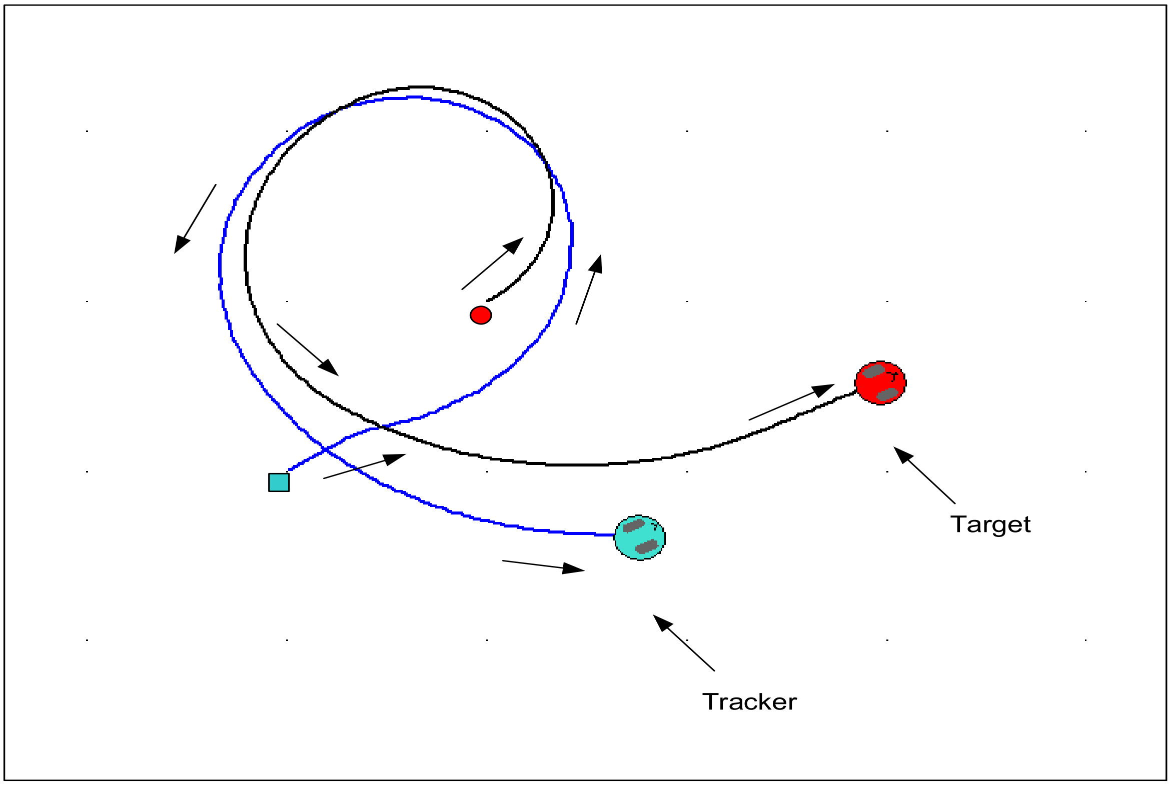


Figure 6.7: single target tracking using supervised learning EKM

parameters can be assumed initially , which will be changed during unsupervised learning. Selecting the initial parameters is very challenging, because it will affect the speed of convergence and the accuracy of the tracking algorithm. In this thesis, the parameters of the Extended Kohonen Map are initially selected from the supervised learning stage (mentioned above). After that, the single target tracking algorithm mentioned in chapter 5 is used to adjust these parameters, while the mobile sensor is trying to track the target (online training). At each training cycle, the weights of the winning neuron and its neighbors are modified to be closer to the sensory data (which are the angle and distance from the tracker). The input weights of the Extended Kohonen Map are updated towards the actual displacement of the tracker and the output weights are also updated. Thus, there will be a map between the actual displacement of the tracker and its motor commands. After the extended Kohonen Map self organization has converged, the neurons will stabilize in a state such that the input weights represent the displacements traveled by the tracker and the output weights represent the corresponding motor commands that produce these displacements. For any winning neuron ne, given the sensory input $x_p = w_{ne}$, the neuron will produce motor commands y_p which yields a desired displacement $v = w_{ne}$. For sensory input $x_p \neq w_{ne}$ but close to w_{ne}, the motor commands output produced by the winning neuron will still yield the corresponding displacement. Therefore, the more neurons in the Extended Kohonen Map structure, the finer and smoother the motion of the tracker.

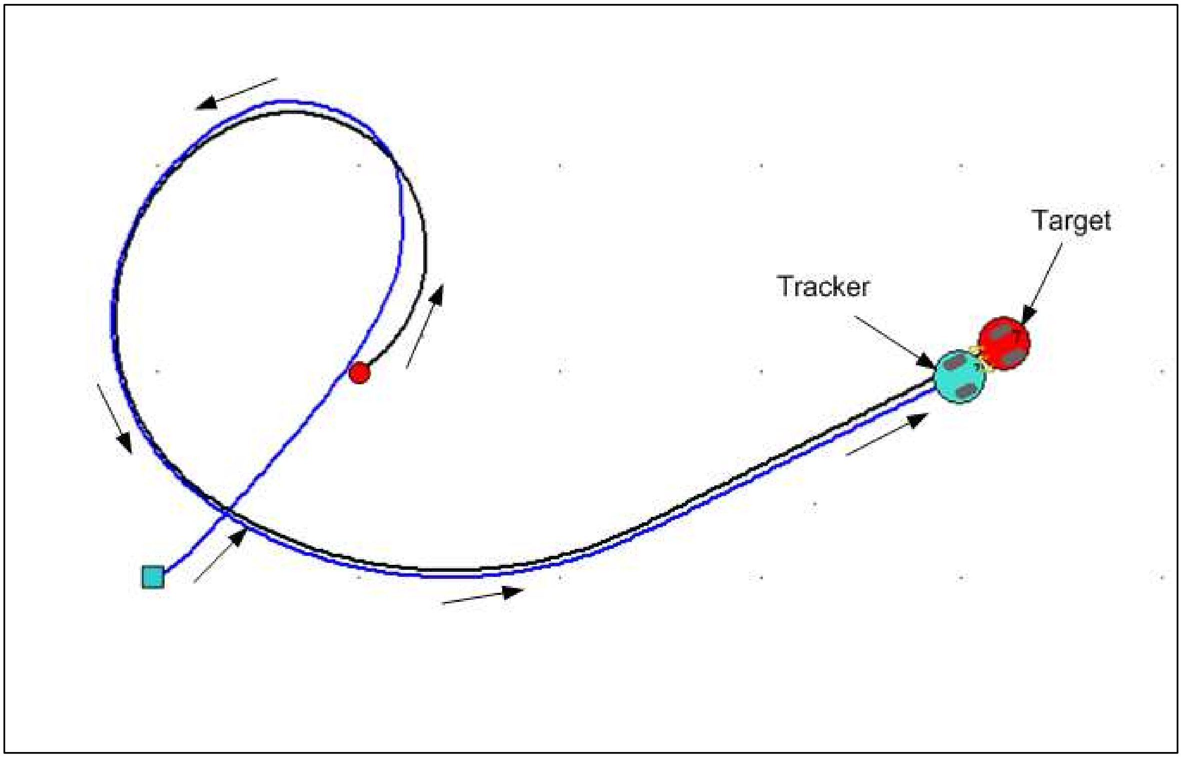


Figure 6.8: single target tracking using unsupervised learning EKM

Fig. 6.8 shows tracking a moving target using unsupervised learning EKM. The input-outputs weights of the unsupervised EKM are initialized using the final weights (after learning) of the supervised EKM. The small circle and square are the initial positions of the target and the tracker respectively. The indicated arrows shows the direction of movement of both the target and the tracker. As seen from the trajectories of the target and the tracker, the unsupervised learning EKM has provided a good tracking performance in terms of accuracy and tracking speed compared to the supervised learning EKM performance shown in Fig. 6.7.

6.2.3 Cooperative Multi-target Tracking Simulations

The proposed multi-target tracking algorithm is analyzed in various configurations. The environment was designed to be a 6m x 9m rectangular shape. The sensors were initially deployed in the environment based on the task allocation algorithm that assigned a set of mobile sensors to a set of vantage points as described in chapter 4. Sensor sensing range was set to be 1.5m. The cooperative multi target algorithm performance is judged by two parameters: the average coverage AC (eq. 5.2) over time and the average energy saving over time AE (eq. 6.1).

$$AE = \sum_{i=1}^{t_e} \frac{st}{n} * \frac{1}{t_e} * 100 \tag{6.1}$$

where st is the number of sleeping trackers, n is the total number of the trackers, and t_e is the execution time of the algorithm.

The energy saving can be measured by considering the number of *busy* and *sleeping* trackers over time. In order to do that, three different cases are analyzed. In the first case, the number of mobile sensors n is equal to the number of targets m, and in the second and third cases, the number of sensors is 3/4 and 1/2 the number of targets respectively. In each case extensive simulations have been done by changing the locations of trackers and targets, the speed of targets and trackers, the direction of targets and trackers,and the number of clusters. The average coverage and average energy saving have been recorded for around 25 trials in each case. The average coverage and the average energy saving for all the cases are shown in Fig. 6.9 and Fig. 6.10 respectively. Each point shows the average coverage or the average energy saving obtained in 25 simulated runs.

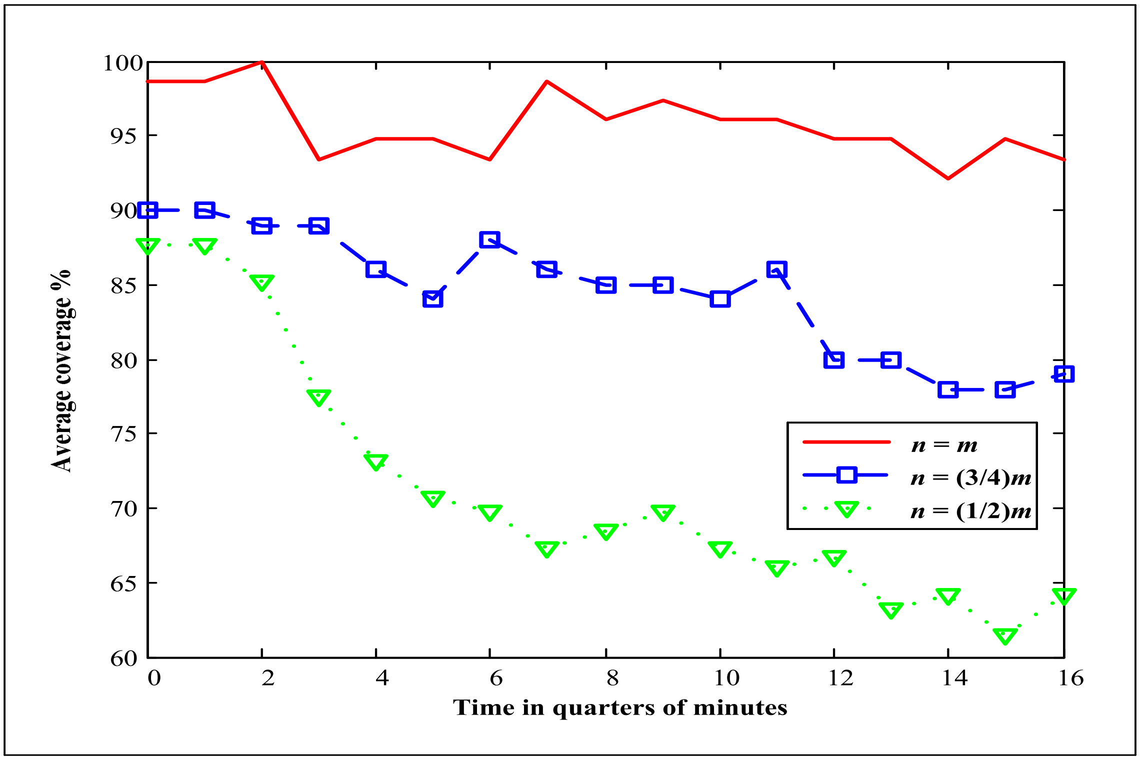


Figure 6.9: comparison between EKM average coverage for different n and m ratios

As shown in Figs. 6.9 and 6.10, the average coverage and the average energy saving percentage decrease with time. This was expected because the clustered targets will spread in the environment over time and which will require more trackers to track them, which decrease the energy savings. Also, the average coverage will decrease especially when the number of targets is more than the number of trackers ($n = (3/4)m$, and $n = (1/2)m$).

The performance of the proposed approach is evaluated by comparing it with four different approaches with different observation policies: (1)Local approach [123], (2)A-

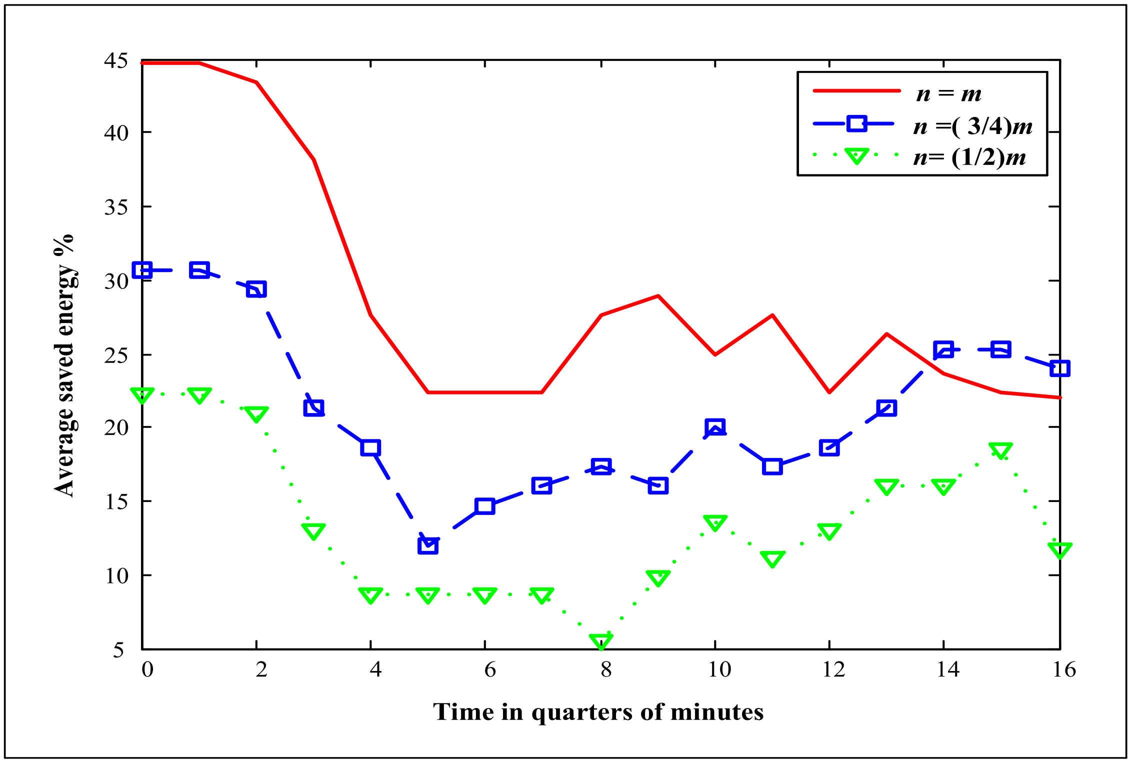


Figure 6.10: comparison between EKM average energy saving for different n and m ratios

CMOMMT approach [123], (3)B-CMOMMT approach [157], and (4)Modified B-CMOMMT [49]. The Local approach controls the tracker motion by computing the summation of the attractive forces coming from nearby targets and the repulsive forces coming from nearby trackers. The A-CMOMMT approach was studied to determine the effectiveness of weighting the force vectors coming from the nearby targets. Adding the weights to the force vectors in A-CMOMMT causes the trackers to be less attractive to targets that are already observed by other trackers. The B-CMOMMT approach is essentially proposed to overcome some problematic situations that may arise in the A-CMOMMT approach. More detailed discussions of these situations can be found in [49, 157]. The most important difference between A-CMOMMT and B-CMOMMT is that the force vectors coming from nearby trackers are weighted as well. The B-CMOMMT approach is improved by introducing more refined techniques for target loss prediction in [49]. The comparison results are shown in Fig. 6.11.

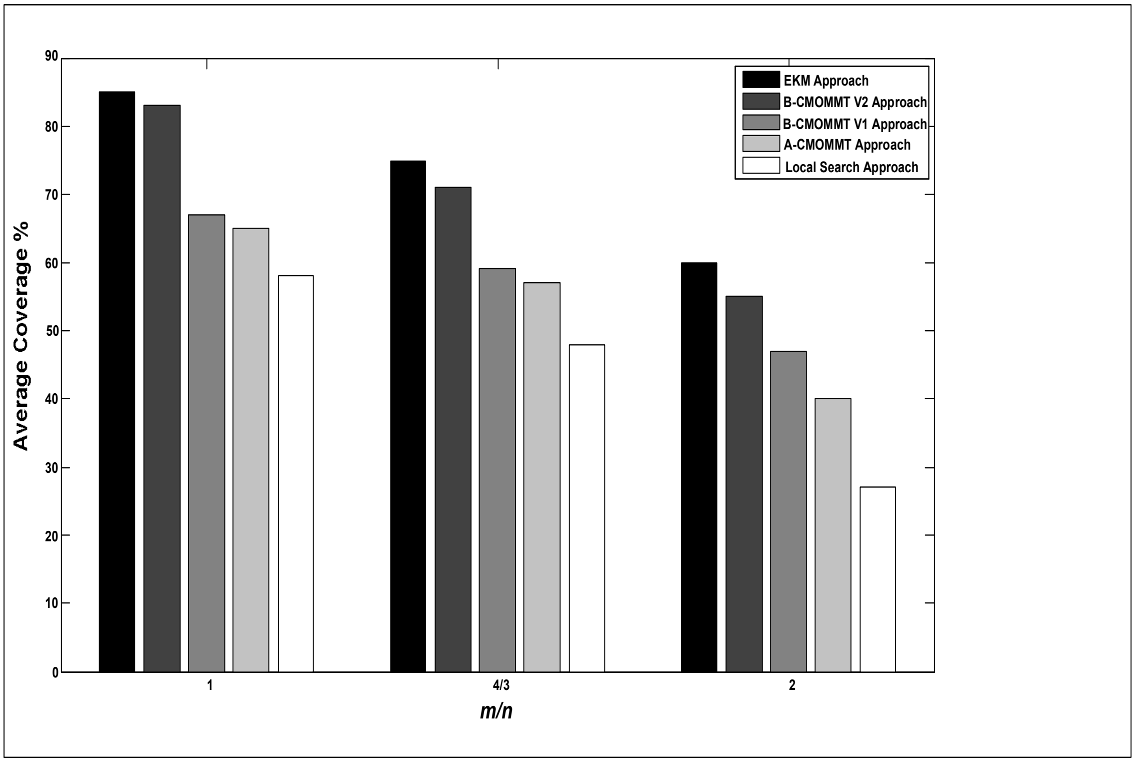


Figure 6.11: comparison between EKM and other approaches

Fig. 6.11 summarizes the results of this comparison with different m/n ratios (every point is the average of 25 simulated runs). These results show nearly equivalent performance of our proposed algorithm and the improved B-CMOMMT approach for small m/n ratios, but also the superior performance of our target tracking approach for larger m/n ratios. Although, we did not prove that the simulation results would hold for larger m/n ratios, it is expected that results similar to what we did will continue for larger m/n ratios.

More interesting is the detailed comparative performance of the proposed approach with the improved B-CMOMMT when m/n equal to one. As shown in Fig. 6.12, the proposed algorithm provides a good coverage performance compared to the B-CMOMMT algorithm. Furthermore, the proposed algorithm provides average energy savings between

8% and 45%, compared to 0% energy saving in the CMOMMT algorithm as shown in table 6.4. The CMOMMT algorithm gives 0% energy saving because all trackers are always busy, even if there are no targets to track (the trackers explore the environment searching for targets even if there are no targets to track).

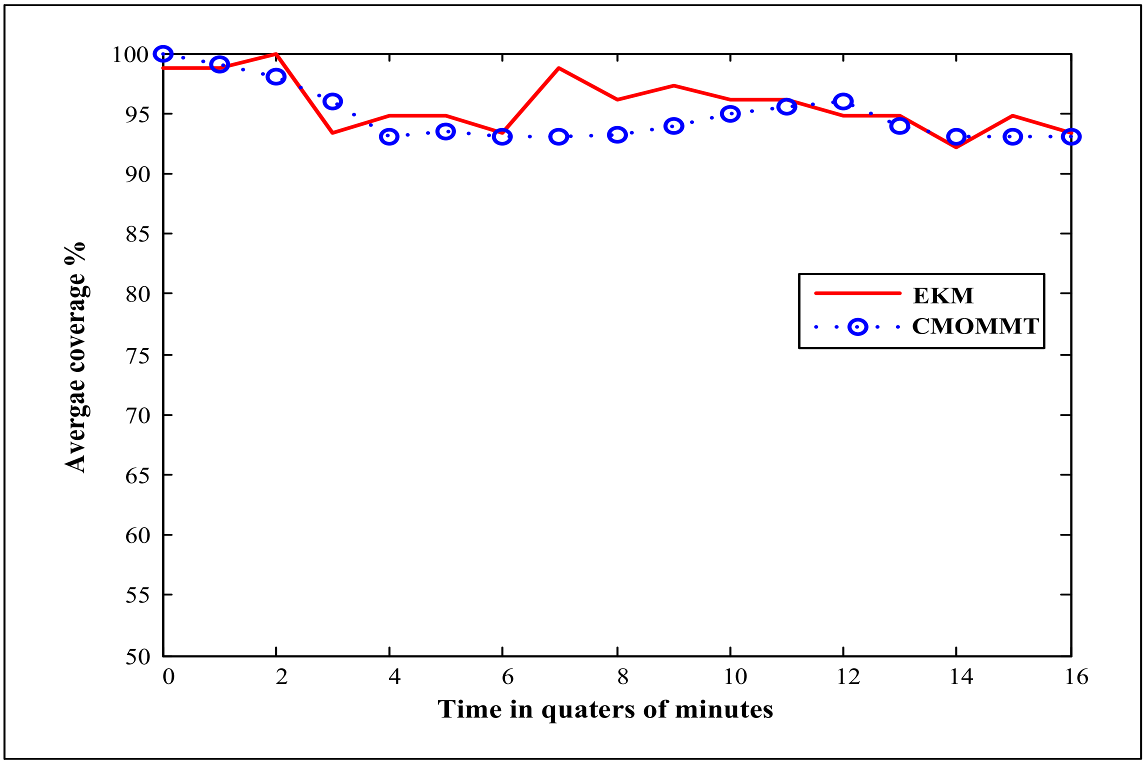


Figure 6.12: comparison between EKM and CMOMMT average coverage for $n = m$

Table 6.4: Energy saving comparison

Case	Energy saving in EKM	Energy saving in CMOMMT
n=m	20-45%	0%
n=(3/4)m	12-31%	0%
n=(1/2)m	8-23%	0%

An example of the indoor surveillance scenario is shown in Fig. 6.13. As shown, the mobile sensors (red ones) are used to survey the six areas of Waterloo airport using the proposed task allocation algorithm. The mobile sensors in this scenario are not allowed to pass from one AOI to another except through the entrances of each AOI. In other words,, the implemented SSP algorithm which is used to find the order of visiting the vantage points considers moving from AOI to another through the entrances only. After surveying the areas, the mobile sensors track the center of gravity of the detected targets (blue ones) using the proposed tracking algorithm. The green lines indicate that the blue targets are detected by the mobile sensors using the fiducial finder detection algorithm. The cones shown in the figure indicate the sonars the mobile sensors are using for obstacle avoidance.

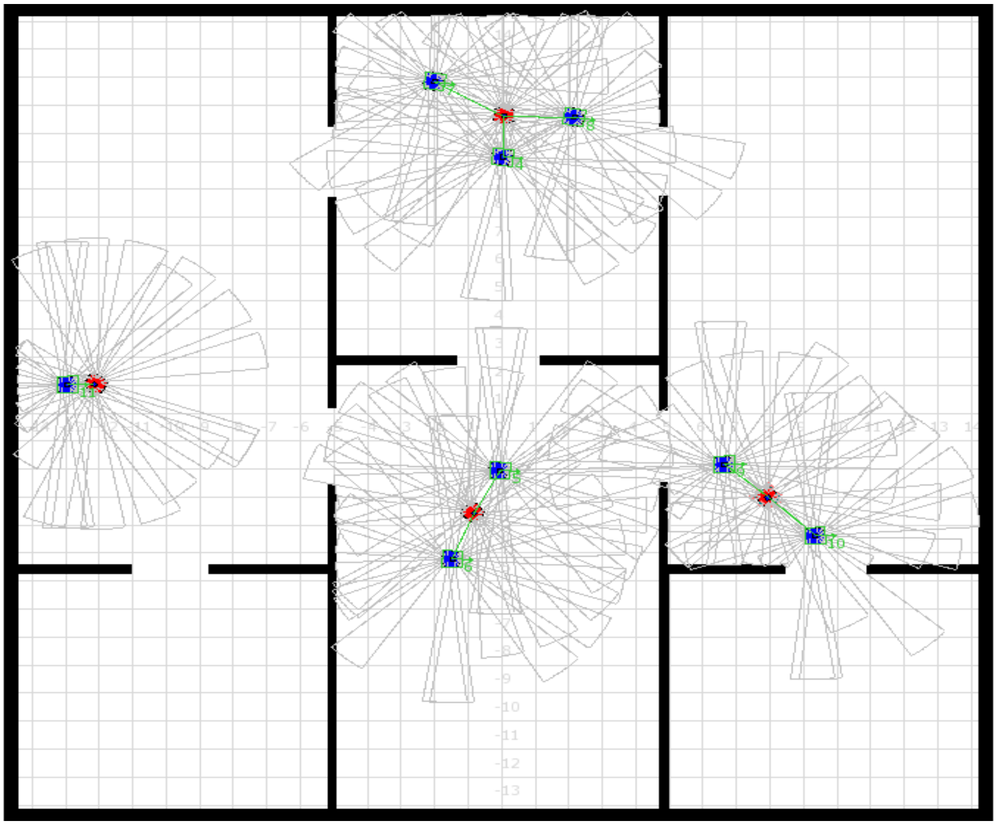


Figure 6.13: Waterloo airport Surveillance

Finally, Fig. 6.14 shows the mission fulfillment time for the Waterloo airport as the number of available mobile sensors is varied. The mission fulfillment time represents the total required time to reposition the mobile sensors according to the designed vantage points, search the areas of interest, and track the detected targets in these areas of interest. As seen, the mission fulfillment time decreases as the number of mobile sensors increases.

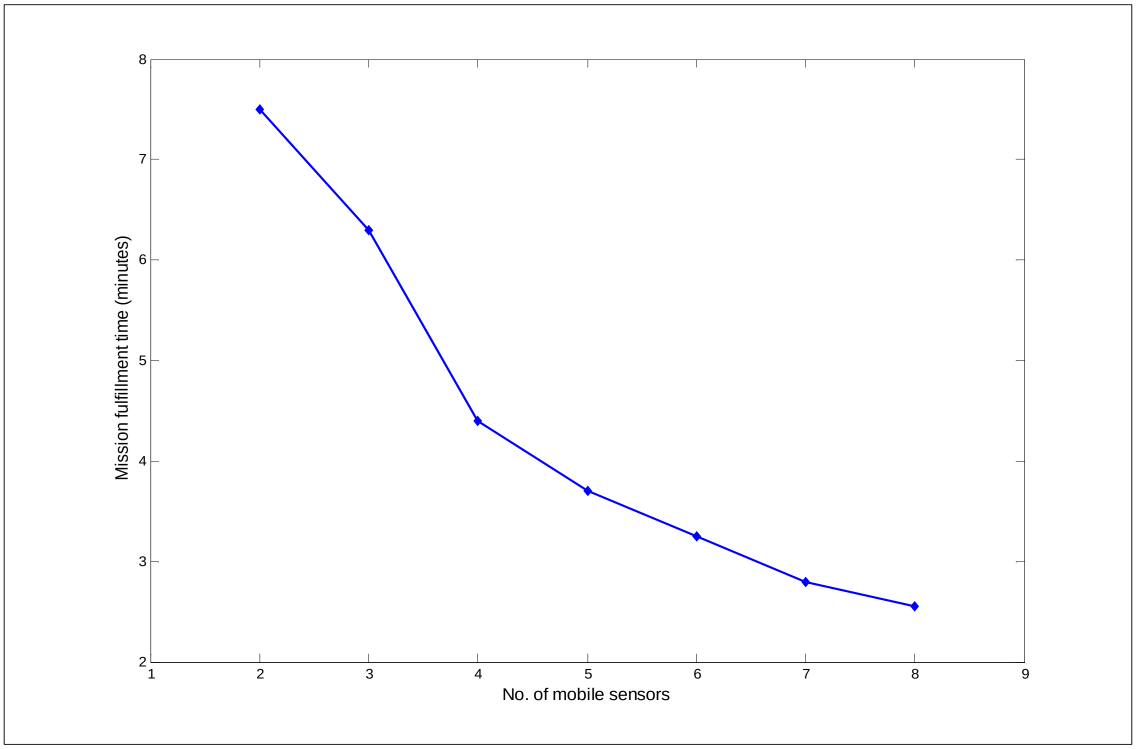


Figure 6.14: Mission fulfillment time for Waterloo airport surveillance

6.3 Conclusion

In this chapter, the simulations and results for complex task allocation are presented. Two scenarios were introduced: outdoor, and indoor. The results of the conducted experiments showed that hierarchical dynamic tree task allocation outperforms all other techniques, especially in complex surveillance operations. Also, the simulation results for multi-sensor target tracking have been presented. Two versions of an Extended Kohonen Map (EKM) are presented; supervised, and unsupervised. The results for cooperative multi target tracking have been presented as well. It has been shown that the unsupervised EKM gives a better performance than the supervised Extended Kohonen Map in terms of the tracking accuracy and speed of convergence. The proposed tracking algorithm has shown also a good coverage performance compared to the existing literature as wellas having the ability to outperform the other existing approaches in terms of energy saving.

Chapter 7

Conclusion

7.1 Conclusion

In recent years, mobile surveillance systems research has received an increasing amount of attention from researchers in academia, government laboratories and industry [34, 96,101]. This research activity has borne some fruit in tackling some of the challenging problems of mobile surveillance that are still open. These research problems include, but are not limited to: task allocation, mobile sensor deployment, multisensor management, cooperative object detection and tracking, decentralized data fusion, and interoperability and accessibility of system nodes.

Most of the current research in the field of surveillance systems focuses on studying the performance of these systems while tackling some of the previously mentioned challenges, but without consideration of tackling them collectively seeking a unified surveillance framework.

A market-based framework for mobile surveillance systems has been presented in this thesis. The proposed framework capitalizes on the strengths of market economies that enable mobile sensing agents to collectively execute complex tasks efficiently and reliably. Task allocation and cooperative target-tracking have been studied in this thesis using the proposed framework as two challenging problems of mobile surveillance systems. These challenges are addressed individually and collectively.

For the task allocation problem, centralized and hierarchical complex task allocation algorithms with fixed and dynamic trees have been presented. Both breadth first, and depth first search mechanisms have been used to search the proposed task trees. It has been shown through extensive simulation experiments using Player/Stage simulator that hierarchical dynamic tree task allocation outperforms other existing techniques, especially in complex surveillance operations where a large number of mobile sensors is used to scan a large number of areas.

A cooperative multi-target tracking methodology has also been proposed based on the proposed framework. Two kinds of target tracking algorithms have been designed and implemented: supervised, and unsupervised. The proposed target tracking algorithms are based on an Extended Kohonen neural network. Also, a hybrid clustering technique has been proposed to cluster the targets in the given environment to minimize the number of active sensors while tracking the moving targets. In order to coordinate the performance of the mobile sensors during tracking, a cueing/handoff market-based algorithm has been presented.

A set of experiments are designed to evaluate the performance of the developed algorithmic solutions in indoor and outdoor scenarios. Analysis of the experimental results shows that the proposed market-based framework successfully handles complex task allocation and cooperative target detection and tracking in mobile surveillance systems.

7.2 Future Work

While many questions have been answered in this thesis, there are a number of potential research directions that are worth pursuing to build upon this research. Some of these directions are as follows:

1. Although the performance of the proposed surveillance framework has been investigated in both outdoor, and indoor environments, the proposed framework may be developed to be a more generic framework to cope with a variety of complex and dynamic environments.

2. Further research may also investigate the effect of using combinations of stationary and mobile sensors on the performance of the proposed surveillance framework.

3. The proposed task allocation algorithms can be extended to handle constrained and tight tasks. An example for constrained tasks could be two tasks which cannot be done independently, as the same sensor would obviously have to do both of them. Tight tasks cannot be decomposed into further single sensor tasks. In this case, a subgroup of mobile sensors could determine their joint costs and submit joint bids for such type of tasks. Also to ensure cost independence between the sub-teams, the proposed framework should be extended to include the constraint that the sub-teams being awarded tight tasks are disjoint.

4. Another extension that may be incorporated into the proposed surveillance framework is the minimization of team makespan. This can be done by changing the global objective function to minimize the maximum cost incurred by any mobile sensor while optimizing overall mission time.

5. The proposed framework may also be developed to tackle the other surveillance challenges like situation awareness and sensor management.

6. The proposed framework may also be developed to tackle extended type of assignment. This will require developing some scheduling algorithms to deal with this type of assignment.

7. Test and evaluate the performance of the proposed setup in a real testbed that realistically simulates surveillance systems. This testbed can be set up by integrating a set of small mobile robots (Khepera III for example) with a wireless sensor network, RFID and wireless cameras.

Bibliography

[1] A. Khamis F. O. Karray B. Khaleghi, S. Navabzadeh. Multisensor data fusion: Antecedents and directions. In *International Conference on Signals, Circuits and Systems (SCS'09)*, Djerba, Tunisia, 2009.

[2] A. Giret and V. Botti. Holons and agents. *International Journal of Intelligent of Manufacturing*, 15(5):645–659, Oct. 2004.

[3] E. M. Petriu, T. E. Whalen, R. Abielmona, and A. Stewart. A robotic sensory agents: a new generation of intelligent agents for complex environment monitoring. *IEEE Instrumentation & Measurement Magazine*, Sep. 2004.

[4] M. D. Bakhtari, M. Eskandari, and B. Benhabib. A multi-sensor surveillance system for active-vision based object localization. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 1013–1018, Washington, D.C., October 2003.

[5] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. D. Nayar, H. Aghazarian, M Garrett A. J. Ganino, S S. Joshi, and P. S. Schenker. Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Trans. on Syst., Man, and Cybern. A, Syst., and Humans*, 33(5):550–559, Sep. 2003.

[6] S. Phoha, E. M. Peluso, and R. L. Culver. A high-fidelity ocean sampling mobile network (samon) simulator testbed for evaluating intelligent control of unmanned underwater vehicles. *IEEE journal of Oceanic Eng.*, 26(4):646–653, Oct. 2001.

[7] J. Casper and R. R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Trans. on Syst., Man, and Cybern. B, Cybernetics*, 33(3):367–385, Jun. 2003.

[8] R. R. Murphy. Human-robot interaction in rescue robotics. *IEEE Trans. on Syst., Man, and Cybern. C, Applications, and Reviews*, 34(2):138–152, May 2004.

[9] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. on Syst., Man, and Cybern. B, Cybernetics*, 35(3):397–408, Jun. 2005.

[10] X. Wang, S. Wang, and D. Bi. Distributed visual-target-surveillance system in wireless sensor networks. *IEEE Trans. on Syst., Man, and Cybern. B, Cybernetics*, 39(5):1134–1146, Oct. 2009.

[11] F. M. Delle Fave, S.Canu, and L. Iocchi. Multi-objective multi-robot surveillance. In *Proceedings of the 4th International Conference on Autonomous Robots and Agents*, pages 68–73, Wellington, New Zealand, Feb. 2009.

[12] A. Elmogy, F. Karray, and A. Khamis. Auction-based consensus mechanism for cooperative tracking in multi-sensor surveillance systems. In *Proceedings of 4th Int. Conf. in Humanoid, Nanotechnology, Information Technology, Communication Control, Environment and Management (HNICEM)*, pages 149–158, Manila, Philippines, March 2009.

[13] H. D. White. Sensor models and multisensor integration. *Journal of Robotics Research*, 7(6):97–113, 1998.

[14] A. Benaskeur, A. Khamis, and H. Irandoust. Multisensor cooperation in military surveillance systems. In *Proceedings of International Conference on Signals, Circuits and Systems (SCS'09)*, Djerba, Tunisia, Nov. 2009.

[15] Y. Q. Miao. A study of mobility models in mobile surveillance systems. Master's thesis, University of Waterloo, Canada, April 2010.

[16] B. P. Gerkey and M. J. Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, Oct. 2002.

[17] M. B. Dias and A. Stentz. A market approach to multi-robot coordination. Technical Report CMU-RI -TR-01-26, Carnegie Mellon University, Pittsburgh, PA, March 2002.

[18] M. B. Dias. *TraderBots: a New Paradigm for Robust and Efficient Multi-robot Coordination in Dynamic Environments*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2004.

[19] R. Zlot and A. Stentz. Market-based multirobot coordination for complex tasks. *Int. Journal of Robotics Research*, 25(1):73–101, 2006.

[20] K. Lerman, C. Jones, A. Galstyan, and J. Mataric. Analysis of dynamic task allocation in multi robot systems. *Int. Journal of Robotics Research*, 25(3):225–241, 2006.

[21] F. Tang and L. Parker. A complete methodology for generating multi-robot task solutions using asymtre-d and market-based task allocation. In *Proceedings of IEEE Int. Conf. On Robot. Autom.*, pages 3351–3358, Roma, Italy, Jul. 2000.

[22] R. M. Zlot. *An Auction-Based Approach to Complex Task Allocation for Multirobot Teams*. PhD thesis, Pittsburgh, Pennsylvania, October 2006.

[23] G. L. Foresti, C. S. Regazzoni, and P. K. Varshney. *Multisensor Surveillance Systems: The Fusion Perspective*. Kluwer Academic Publishers, 2003.

[24] M. Valera and S. Velastin. Intelligent distributed surveillance systems:a review. *IEE Proceedings on Vision, Image and Signal Processing*, 152(2):192–204, April 2005.

[25] Multimodal-surveillance system for security-related applications (muses_secret) project. http://129.97.86.45:88/redmine/wiki/muses, Last accessed: March 29, 2010.

[26] J. M. Ferryman, S. J. Maybank, and A. D. Worrall. Visual surveillance for moving vehicles. *International Journal of Computational Vision*, 37(2):187–197, June 2000.

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

[28] B. Lo, S.A. Velastin, M.A. Vicencio-Silva, and J. Sun. *Military Transformation: Intelligence, Surveillance, and Reconnaissance.* Distributed Surveillance Systems (IDSS), fvrier, 2003.

[29] G. J. Chizek. *Military Transformation: Intelligence, Surveillance, and Reconnaissance*. Congressional Research Service, Washington DC, USA, 2003.

[30] S. Rathinam, R. Sengupta, and S. Darbha. A resource allocation algorithm for multivehicle systems with nonholonomic constraints. *IEEE Trans. on Automation Science and Engineering*, 4(1):98–104, Jan. 2007.

[31] J. Pavna, J. G. Sanza, A. F. Caballerob, and J. V. Jimnezc. Development of intelligent multisensor surveillance systems with agents. *Robotics and Autonomous Systems*, 55(12):892–903, Dec. 2007.

[32] S. A. Velasin and P. Remagnino. Intelligent distributed video surveillance systems. In *IEE Professional Applications of Computing Series 5*, 2006.

[33] J. M. Lpez, J. G. Herrero, F. J. Rodrguez, and J. R. Corredera. Fuzzy reasoning in a multiagent system of surveillance sensors to manage cooperatively the sensor-to-task assignment problem. *Applied Artificial Intelligence*, 18(8):671–711, 2004.

[34] A. Elmogy, F. Karray, and A. Khamis. Auction-based consensus mechanism for cooperative tracking in multisensor surveillance systems. *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, 14(1):13–19, 2010.

[35] A. Elmogy, A. Khamis, and F. Karray. Market-based dynamic task allocation in mobile surveillance systems. *Selected to be Published in a Special Issue of Journal of Intelligent and Robotics Systems, Springer*, 2010.

[36] R. F. Haines. *Night Flying.* Blue Ridge Summit, PA: TAB Books, 1992.

[37] N. A. Stanton, P. G. Chambers, and J. Piggott. Situational awareness and safety. *Safety Science*, 39, 2001.

[38] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[39] What is situation awareness? http://www.satechnologies.com/situation awareness, 2010.

[40] A. Benaskeur and H. Irandoust. Holonic approach for control and coordiantion of distributed sensors. *Defence R&D Canada - Valcartier*, 2008.

[41] L. Hodge and M. Kamel. An agent-based approach to multisensor coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 33(4):648–661, 2008.

[42] G. W. Ng and K. H. Ng. Sensor management what, why and how. *Information Fusion*, pages 67–75, December 2000.

[43] R.V. Denton, E. I. Alcaraz, J. Llinas, , and K. J. Hintz. Towards modern sensor management systems, science of command and control. *Part 3 Coping with Change, AFCEA International Press*, pages 119–134, 1994.

[44] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[45] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: a market-based framework for complex tight coordination in multi-robot teams. In *Proceedings of IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1170–1177, 2005.

[46] S. C. Botelho and R. Alami. M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement. In *Proceedings of IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1234–1239, 1999.

[47] S. Sariel-Tala, T. R. Balch, and N. Erdogan. Multiple traveling robot problem: a solution based on dynamic task selection and robust execution. *IEEE/ASME Trans. on Mechatronics*, 14(2):198–206, Apr. 2009.

[48] T. Shima, S. Rasmussen, and D. Gross. Assigning micro uavs to task tours in an urban terrain. *IEEE Trans. on Control Syst. Technology*, 15(4):601–612, Jul. 2007.

[49] A. Kolling and S. Carpin. Cooperative observation of multiple moving targets: an algorithm and its formalization. *International Journal of Robotics Research*, 26(9):935–953, 2007.

[50] J. R. Spletzer and C. J. Tayor. Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research*, 22(1):7–20, Jaan. 2003.

[51] B. Jung. *Cooperative Target Tracking using Mobile Robots.* PhD thesis, Univ. of Southern California, California, May 2005.

[52] B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(4):281–316, 2004.

[53] K. M. Carley and L. Gasser. *Computational Organization Theory.* Lawrence Earlbaum Associates, Hillsdale, NJ, 1994.

[54] P. Wall. *Centralized versus Decentralized Information Systems in Organizations.*

[55] B. L. Brumitt and A. Stentz. Grammps: a generalized mission planner for multiple mobile robots in unstructured environments. In *Proceedings of IEEE Int. Conf. on Robot. Autom.*, pages 1564–1571, Leuven, Belgium, May 1998.

[56] Y. Khalifa and E. Okoene. A centralized evolutionary target tracking system. In *Proceedings of the ASEE New England Section 2006 Annual Conference*, England, 2006.

[57] M. Nieminen, T. Rty, and M. Lindholm. Multi-sensor logical decision making in the single location surveillance point system. In *Proceedings of 2009 Fourth International Conference on Systems*, Gosier, Guadeloupe, France, 2009.

[58] B. S. Rao, H. F. DurrantWhyte, and J. A. Sheen. A fully decentralized multisensor system for tracking and surveillance. *International Journal Robotics Research*, 12(1):20–44, 1993.

[59] X. Li, G. Chen, E. Blasch, J. Patrick, C. Yang, and I. Kadar. Multi-sensor management for data fusion in target tracking. volume 7336, page 73360Y. SPIE, 2009.

[60] S. Oh and S. Sastry. A hierarchical multiple-target tracking algorithm for sensor networks. In *Proceedings of the International Conference on Robotics and Automation*, 2005.

[61] Z. Qingguang, C. Yanling, and L. Juan. A lightweight key management protocol for hierarchical sensor networks. *International Conference on Parallel and Distributed Computing Applications and Technologies*, pages 379–382, 2006.

[62] R. W. Brennan. Holonic and multi-agent systems in industry. *The Knowledge Engineering Review*, 16(4):375–381, Dec. 2001.

[63] Gaud N. Hilaire V. Rodriguez, S. and Koukam. Modeling holonic systems with an organizational approach. *Proceedings of EUMAS*, 2006.

[64] A. BENASKEUR, P. MCGUIRE, R. BRENNAN, G. LIGGINS, and P. WOJCIK. A distributed intelligent tactical sensor management system. *International Journal of Intelligent Control and Systems*, 12(2):97–106, June 2007.

[65] A. Koukam E. F. Tagne, F. Lauri and E. Tonye. The coverage problem in wireless sensor networks by holonic multi-agent approach. *International Journal of Computing and ICT Research*, 3(1):32–41, 2009.

[66] M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and xml data. In *proceed. Of Arena*, 2003.

[67] J. H. Smith. Some observations on the concepts of information theoretic entropy and randomness. *Entropy 2001*, 3(1):1–11, 2001.

[68] A. Andrea and S. Salah. An information theoretic approach to modeling voting and elections. In *Paper presented at the Annual Meeting of the Western Political Science Association*, Hyatt Regency Albuquerque, Albuquerque, New Mexico, March 2006.

[69] J. L. Williams. *Information Theoretic Sensor Management*. PhD thesis, MIT, 2007.

[70] M. T. J. Spaan and P. U. Lima. A decision-theoretic approach to dynamic sensor

selection in camera networks. In *Proceedings of 2009 American Control Conference*, pages 279–304, St. Louis, MO, USA, 2009.

[71] Y. Lan, G. Yan, and Z. Lin. A hybrid control approach to cooperative target tracking with multiple mobile robots. In *Proceedings of 2009 American Control Conference*, pages 2624–2629, St. Louis, MO, USA, June 2009.

[72] N. Li, H. Zhang, and H. Ouyang. Shape optimization of coronary artery stent based on a parametric model. *Finite Elements in Analysis and Design*, 45(6-7):468–475, May 2009.

[73] P. J. Shea, K. Alexander, and J. Peterson. Group tracking using genetic algorithms. In *International Society Information Fusion*, pages 680–687, San Diego, USA, 2003.

[74] L. Chen Y. F. Che Y. Yu, L. Guo. Ant colony optimization for sensor management. In *Proceedings of IEEE International Conference on Control and Automation*, pages 576–580, Guangzhou, China, May 2007.

[75] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. on Computers*, C-29(12), 1980.

[76] J. S. Banks, J. O. Leryard, and D. Porter. Allocating uncertain and unresponsive resources: an experimental approach. *The RAND Journal of Economics*, 20(1):1–25, 1989.

[77] T. Mullen, V. Avasarala, and D. L. Hall. Customer-driven sensor management. *IEEE Intelligent Systems*, 21(2):41–49, Mar/April 2006.

[78] W. Z. Yan, Y. J. Peng, and W. Shu. Market-based adaptive task scheduling for sensor networks. In *Proceedings of Wireless Communications, Networking, and Mobile Computing, WiCOM 2006. International Conference*, Sep. 2006.

[79] S. Martinez-Jaramillo and E. P. K. Tsang. An heterogeneous, endogenous and co-evolutionary gp-based financial market. *IEEE Trans. on Evol. Comput.*, 13(1):33–55, 2009.

[80] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multi-robot system. In *Proceedings of 6th Int. Conf. on Intelligent Autonomous Syst. (IAS-6)*, pages 115–122, Jul. 2000.

[81] T. Sandholm. *Negotiation Among self Interested Computationally Limited Agents.* PhD thesis, University of Massachusetts Amherst, 1996.

[82] A. Stentz and M. B. Dias. A free market architecture for coordinating multiple robots. Technical Report CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1999.

[83] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations.* Cambridge University Press, Cambridge, UK, 2009.

[84] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1957–1962, Oct. 2003.

[85] M. Reck. Types of electronic auctions. In *Proceedings of the International Conference on Information and Communications Technologies in Tourism*, 1994.

[86] J. Vidal. Fundamentals of multiagent systems with netlogo examples. WWW page.

[87] P. Cramton, Y. Shoham, and R. Steinberg (Eds). *Combinatorial Auctions.* MIT Press, Cambridge, MA,USA, 2006.

[88] J. Lee, S. Lee, , and S. Lee. A framework of winner determination algorithms for internet auctions. *Springer-Verlag Berlin Heidelberg*, 2713, 2003.

[89] T. W. Sandholm. *Optimal Winner Determination Algorithms .In P. Cramton, Y. Shoham, and R. Steinberg, editors, Combinatorial Auctions.* MIT Press, Cambridge, MA,USA, 2006.

[90] T. Sandholm and V. Lesser. Leveled commitment contracting: a backtracking instrument for multiagent systems. *AI Magazine*, 23(3):89–100, 2002.

[91] M. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: a survey and analysis. Technical Report CMU-RI-TR-05-13, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Apr. 2005.

[92] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain. Auction-based multirobot routing. In *Proceedings of Robotics: Science and Systems (RSS))*, pages 343–350, Cambridge, USA, 2005.

[93] C. Tovey, M. G. Lagoudakis, S. Jain, and S. Koenig. The generation of bidding rules for auction-based robot coordination. In *Proceedings of the 3rd International Multi-Robot Systems Workshop*, Naval Research Laboratory, USA, March 2005.

[94] M. B. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In *Proceedings of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2714–2720, Lausanne, Switzerland, March 2005.

[95] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market-economy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[96] B. P. Gerkey and M. J. Mataric. A framework for studying multi-robot task allocation. 2003.

[97] D. Gale. *The Theory of Linear Economic Models*. McGraw-Hill Book Company, New York, USA, 1960.

[98] G. P Gerkey and M. J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. Journal of Robotics Research*, 23(9):939–954, 2004.

[99] B. P. Gerkey and M. J. Mataric. Sold!: Auction methods for multi-robot control. *IEEE Trans. Robot. Autom. (Special Issue on Multi-Robot Systems)*, 18(5):758–768, Oct. 2002.

[100] L. Parker. Alliance: an architecture for fault-tolerant multi-robot cooperation. *IEEE Trans. on Robot. and Autom.*, 14(2):220–240, May 2004.

[101] F. Tang and S. Saha. An anytime winner determination algorithm for time-extended multi-robot task allocation. In *Int. Conf. on Automation, Robotics, and Control Syst.*, pages 123–130, 2008.

[102] M. B. Dias, M. Zinck, R. Zlot, and A. Stentz. Robust multi-robot coordination in dynamic environments. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3435–3442, 2004.

[103] B. Zaki and M. Gammoudi. Decentralized method for complex task allocation in massive mas. In *Proceedings of Intelligent Syst. and Automation: 1st Mediterranean Conf. on Intelligent Systems and Automation (CISA 08)*, pages 287–293, 2008.

[104] E. Manisterski, E. David, E. Kraus, and N. Jennings. Forming efficient agent groups for completing complex tasks. In *Proceedings of 5th Int. Conf. on Autonomous Agents and Multi-Agent Syst.*, pages 834–841, Hakodate, Japan, 2006.

[105] C. J. Ortiz, R. Vincent, and B. Morisset. Task inference and distributed task management in the centibots robotic system. In *Proceedings of 4th Int. Joint Conf. on*

Autonomous Agents and Multi-Agent Syst., pages 860–867, Utrecht University, the Netherlands, Jul. 2005.

[106] R. Aylett and D. Barnes. A multi-robot architecture for planetary rovers. In *Proceedings of 5th ESA workshop on Advanced Space Technologies for Robot. Autom.*, Noordwick, the Netherland, Dec. 1998.

[107] P. Caloud, W. Choi, J. C. Latombe, C. L. Pape, and M. Yim. Indoor automation with many mobile robots. In *Proceedings of IEEE Int. Workshop on Intelligent Robots and Syst.(IROS)*, 1990.

[108] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Communication and coordination among heterogeneous mid-size players: Art99. *in*, 2001.

[109] R. Simmons, D. Apfelbaum, D. Fox, R. Goldman, K. Haigh, D. Musliner, M. Pelican, and S. Thrun. Coordinated deployment of multiple heterogeneous robots. 2000.

[110] X. F. Xie and J. Liu. Multiagent optimization system for solving the traveling salesman problem (tsp). *IEEE Trans. on Syst., Man, and Cybern. B, Cybernetics*, 39(2):489–502, 2009.

[111] H. K. Tsai, J. M. Yang, Y. F. Tsai, and C. Y. Kao. An evolutionary algorithm for large traveling salesman problems. *IEEE Trans. on Syst., Man, and Cybern. B, Cybernetics*, 34(4):1718–1729, 2004.

[112] C. Wurll, D. Henrich, and H. Worn. Multi-goal path planning for industrial robots. In *Proceedings of International Conference on Robotics and Application (RA99)*, Santa Barbara, USA, Oct. 1999.

[113] X. Zheng, S. Koenig, and C. Tovey. Improving sequential single-item auctions. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2238–2244, Beijing, China, 2006.

[114] M. H. Rothkopf, A. Pekec, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 4(8):1131–1147, 1998.

[115] K. Hsiang, W. Leow, and M. Ang. Autonomic mobile sensor network with self-coordinated task allocation and execution. *IEEE Trans. On SMC- Part C*, 36(3):315–327, May 2006.

[116] D. Liu and L. C. Fu. Target tracking in an environment of nearly stationary and biased clutter. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1358–1363, Maui, Hawaii, USA, Oct. 2001.

[117] D. Carevic. Kalman filter-based approach to target detection and target background separation in ground-penetrating radar data. In *Proceedings of SPIE Conf. Detection Remediation Technol. Mines, Mine like Targets IV*, pages 1284–1288, 1999.

[118] Z. Cai and Z. Duan. Adaptive evolutionary particle filter based object tracking with occlusion handling. In *Proceedings of Fifth International Conference on Natural Computation*, Tianjian, China, Aug. 2009.

[119] P. Arambel, J. Silver, J. Krant, M. Antone, and T. Strat. Multiple-hypothesis tracking of multiple ground targets from aerial video with dynamic sensor control. In *Proceedings of SPIE 5429, (Signal Processing, Sensor Fusion, and Target Recognition XIII)*, pages 23–32, Tianjian, China, Aug. 2004.

[120] Z. Ding, H. Leung, and L. Hong. Decoupling joint probabilistic data association algorithm for multiple target tracking. *IEE Trans. on Radar, Sonar Navigation*, 146(5):251–254, 1999.

[121] Y. Xue and D. Morrell. Target tracking and data fusion using multiple adaptive foveal sensors. In *Proceedings of International Conference on Information Fusion*, July 2003.

[122] F. Amoozegar. Virtual target tracking (vtt) as applied to mobile satellite communication networks. In *Proceedings of SPIE*, volume 3718, pages 562–572, July 2003.

[123] L. E. Parker. Distributed algorithms for multi-robot observation of multiple moving targets. *International Journal of Autonomous Robots*, 12:231–255, 2002.

[124] J. Kang, I. Cohen, and G. Medioni. Continuous multi-views tracking using sensor voting. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 181–186, Florida, USA, Dec. 2002.

[125] A. Biswas, P. Guha, A. Mukerjee, and K.S. Venkatesh. Intrusion detection and tracking with pan-tilt cameras. In *Proceedings of IET International Conference on Visual Information Engineering*, volume 3718, pages 565–571, 2006.

[126] S. Saripalli. Vision-based autonomous landing of an helicopter on a moving target. In *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Chicago, USA, Aug. 2009.

[127] B. Jung and G. S. Sukhatme. A generalized region-based approach for multi-target tracking in outdoor environments. In *Proceedings of IEEE Int. Conf. on Robotics and Automation (ICRA04)*, New Orleans, LA, April 2004.

[128] C. Y. Lee. *Real-time Target Tracking in an Indoor Environment.* PhD thesis, Stanford University, Stanford, May 2002.

[129] H. Ma and B. H. Ng. Distributive target tracking in wireless sensor networks under measurement origin uncertainty. In *Proceedings of 3rd International Conference In Intelligent Sensors, Sensor Networks and Information*, pages 299–304, 2007.

[130] B. Gerkey, R. Vaughan, K. Stoy, A. Howard, G. S. Sukhatme, and M. Mataric. Most valuable player: A robot device for distributed control. In *Proceedings of the*

IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1226–1231, Wailea, Hawaii, Oct. 2001.

[131] L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, Special Issue on Robotics Research at Oak Ridge National Laboratory*, 5(1):5–19, 1999.

[132] W. Zhang and G. Cao. Optimizing tree reconfiguration for mobile target tracking in sensor networks. In *Proceedings of the IEEE International Conference on Computer Communication*, March. 2004.

[133] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford Univ. Press, London, U.K., 1987.

[134] N. Heo and P. Varsheny. Energy efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on SMC- Part A*, 35(1), Jan. 2005.

[135] H. Qi, S. S. Iyengar, and K. Chakrabarty. Distributed sensor fusion-a review of recent research. *J. Franklin Inst.*, 338:655–668, Jan. 2005.

[136] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In *Proceedings of 6th Int. Conf. Distributed Autonomous Robotic Systems*, pages 299–308, Fukuoka, Japan, 2002.

[137] J. Cortes, S. Martnez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions On Robotics and Automation*, 20(2):655–668, April 2004.

[138] S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J. C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proceedings of the 1997 In-*

ternational Conference on Robots and Automation, pages 731–736, Fukuoka, Japan, 1997.

[139] S. M. LaValle, D. Lin, L. J. Guibas, J. C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proceedings of the 1997 International Conference on Robots and Automation*, pages 737–742, Fukuoka, Japan, 1997.

[140] H. R. Everett, G. A. Gilbreath, T. A. Heath-Pastore, and R. T. Laird. Coordinated control of multiple security robots. In *Proceedings of SPIE Mobile Robots VIII*, pages 292–305, 1993.

[141] Y. B. Shalom. *Multitarget- Multisensor Tracking: Application and Advances.* Volume II. Artech House, Norwood, MA, 1992.

[142] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Aug. 2000.

[143] P. Gerbino and M. Ali. A novel local likelihood approach to data fusion in passive target tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, page 2/1 2/4, 1999.

[144] B. Jung and G. S. Sukhatme. Tracking targets using multiple robots: the effect of environment occlusion. *International Journal of Autonomous Robots*, 13(3):192–205, 2002.

[145] D. Streller, K. Dietmayer, and J. Sparbert. Object-tracking in traffic scenes with multi-hypothesis approach using laser range images. In *Proceedings of IEEE 4th International Conference on Intelligent Transportation System*, 2001.

[146] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki. Monitoring an intersection using a network of laser scanners. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systemsm*, Beijing, China, Oct. 2008.

[147] H. Zhao and R. Shibasaki. A novel system for tracking pedestrians using multiple single-row laser range scanners. *IEEE Trans. SMC Part A: Systems and Humans*, 35(2):283–291, March 2005.

[148] B. Kluge, C. Kohler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1683–1688, 2001.

[149] T. Kanungo, D. M. Mount andN. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, July 2002.

[150] K. Hammouda. A comparative study of data clustering techniques. in: Tools of intelligent systems design. In *Course Project, SYDE 625*, University of Waterloo, Waterloo, Canada, 2000.

[151] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 1:281–297, Berkeley, University of California Press, 1967.

[152] T. Kohonen. *Self-Organization and Associative Memory.* Springer Series in Information Sciences, Heidelberg, third edition, 1989.

[153] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps: an Introduction.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, first edition, 1992.

[154] B. Gerkey, R. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of 11th Int. Conf. on Advanced Robotics*, 2003.

[155] R. Vaughan. Massively multi-robot simulation in stage. *Journal of Swarm Intelligence*, 2(2-4):189–208, 2008.

[156] L. M. Gambardella and C. Versino. Learning the visuomotor coordination of a mobile robot by using the invertible kohonen map. In *Proceedings of International Workshop on Artificial Neural Networks: from Natural to Artificial Neural Computation*, pages 1084–1091, 1995.

[157] A. Kolling and S. Carpin. Multi-robot cooperation for surveillance of multiple moving targets - a new behavioral approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1311–1316, 2006.