A DECISION-THEORETIC APPROACH FOR CONTROLLING AND COORDINATING MULTIPLE ACTIVE CAMERAS IN SURVEILLANCE

PRABHU NATARAJAN

NATIONAL UNIVERSITY OF SINGAPORE

2013

A DECISION-THEORETIC APPROACH FOR CONTROLLING AND COORDINATING MULTIPLE ACTIVE CAMERAS IN SURVEILLANCE

PRABHU NATARAJAN

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2013

ACKNOWLEDGMENTS

First and above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully. The realization of this work was only possible due to the several people's help and support. It is my great pleasure to take this opportunity to thank them.

Foremost, I would like to express my gratitude to my advisors Prof. Mohan Kankanhalli and Prof. Bryan Kian Hsiang Low for their timely advice, guidance, patience and motivation during my PhD studies. Without their continuous support and encouragement, it is impossible for me to complete this thesis. Their guidance helped me in all the times of research and writing of this thesis. I would like to thank my committee members, Prof. Roger Zimmermann and Prof. Wei Tsang Ooi, for their encouraging words, thoughtful criticism, and time and attention during busy semesters. I would also like to thank Prof. Tan Tiow Seng and Mr. Chow Chin Ming for lending the Lego robots for my experiments. I also thank Ms. Judy Ng Lee Chin for her timely help to install the PTZ cameras in our lab. The administrative staff at the School of Computing have been supportive and I take this opportunity to thank Ms. Loo Line Fong, Ms. Irene, and Mr. Mark Bartholomeusz in particular for their commitment and responsiveness time and again. I like to thank my department and university for the excellent facilities given for us in the campus.

I would like to thank my former mentor Dr. Pradeep Kumar for his all-time support and encouragement from my undergraduate studies to now. Without him, I would have not pursued this research career. He remains my best role model for a scientist, mentor, entrepreneur, friend and a teacher. My lab mates and friends played a variety of roles ranging from mentors, buddies and critics, at different times. I want to thank my friends Trong Nghia, Karthik, Ruofei, Chen Jie, Xu Nuo, Dr. Ravindra, Dr. Debashis, Dr. Xiangyu, Dr. Lu Zheng, Dr. Harish, Haoyu Bai, Dr. Mukesh, Gan Tian, Yuhui, Suria, Manoranjan, Suchi and many more for their interesting discussions and support.

I thank my well-wisher and best friend Karthikeyan for all his support and prayers. I would like to thank my parents Mr. Natarajan and Mrs. Gomathi, and my brother Babu, for their constant support, motivations and unconditional love. My final, and most heartfelt, acknowledgment must go to my best friend, soul-mate, and wife Thendral Suganya. Her support, encouragement, and companionship has turned my journey through graduate school into a pleasure. For all that, and for being everything I am not, she has my everlasting unconditional love and care.

To my wife & my family

Contents

Abstract

The use of active cameras in surveillance is becoming increasingly popular as they try to meet the demands of capturing high-resolution images/videos of targets in surveillance for face recognition, target identification, forensic video analysis, etc. These active cameras are endowed with pan, tilt, and zoom capabilities, which can be exploited to provide high-quality surveillance. In order to achieve effective, real-time surveillance, an efficient collaborative mechanism is needed to control and coordinate these cameras' actions, which is the focus of this thesis. The central problem in surveillance is to monitor a set of targets with guaranteed image resolution. Controlling and coordinating multiple active cameras to achieve this surveillance task is non-trivial and challenging because: (a) presence of inherent uncertainties in the surveillance environment (target's motion, location, and noisy camera observation); (b) there exist a non-trivial trade-off between number of targets and the resolution of observing these targets; and (c) more importantly, the coordination framework should be scalable with increasing number of targets and cameras.

In this thesis, we formulate a novel decision-theoretic multi-agent planning approach for controlling and coordinating multiple active cameras in surveillance. Our decision-theoretic approach offers advantages of (a) accounting the uncertainties using probabilistic models; (b) the non-trivial trade-off is addressed by coordinating the active cameras' actions to maximize the number of targets with guaranteed resolution; and (c) the scalability in number of targets and cameras is achieved by exploiting the structures and properties that are present in our surveillance problem. We focus on two novel problems in active camera surveillance: (a) maximizing observations of multiple targets (MOMT), i.e., maximizing the number of targets observed in active cameras with guaranteed image resolution; and (b) improving fairness in observation of multiple targets (FOMT), i.e., no target is "starved" of observation by active cameras for long duration of time.

We propose two formal decision-theoretic frameworks (a) Markov Decision Process (MDP) and (b) Partially Observable Markov Decision Process (POMDP) frameworks for coordinating active cameras in surveillance. MDP framework controls active cameras in *fully observable* surveillance environments where the active cameras are supported by one or more wide-view static/fixed cameras to observe the entire surveillance environment at low-resolution. POMDP framework controls active cameras in *partially observable* surveillance environments where it is impractical to observe the entire surveillance environment using static/fixed cameras due to occlusions caused by physical infrastructures. Hence the POMDP framework do not have a complete view of the surveillance environment.

Specifically, we propose (a) MDP frameworks to solve MOMT problem and FOMT problem in*fully observable* surveillance environment; and (b) POMDP framework to solve MOMT problem in *partially observable* surveillance environment. As proven analytically, our MDP and POMDP frameworks incurs time that is linear in number of targets to be observed during surveillance. We have used max-plus algorithm with our MDP framework to improve its scalability in number of cameras for MOMT problem. Empirical evaluation through simulations in realistic surveillance environment reveals that our proposed approach can achieve high-quality surveillance in real time. We also demonstrate our proposed approach with real Axis 214 PTZ cameras to show the practicality of our approach in real world surveillance. Both the simulations and real camera experiments show that our decision-theoretic approach can control and coordinate active cameras efficiently and hence contributes significantly towards improving the active camera surveillance research.

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Surveillance security is becoming a part of the building infrastructures due to recent security threats like the Mumbai terrorist attack and Boston bomb blasts. Central to the problem of surveillance is that of monitoring, tracking, and observing multiple mobile targets of interest distributed over a large-scale obstacle-ridden environment (e.g., airport terminals, railway and subway stations, bus depots, shopping malls, school campuses, military bases, etc.). It is often necessary to acquire high-resolution videos/images of these targets for supporting real-world surveillance applications like activity/intention tracking and recognition [IBM, 2012], biometric analysis like target identification and face recognition [GE, 2009], surveillance video mining [MER, 2009], forensic video analysis/retrieval [IBM, 2012], among others. Traditional surveillance systems consists of large number of fixed/static analogue CCTV (Closed Circuit Television) cameras that are placed to constantly focus at the selected important locations in the buildings like entrance/exit, lobby, etc. Unfortunately, the maximum resolution of these cameras is limited to 720×480 pixels. So, they cannot capture high-resolution images/videos of the targets, especially when the targets are far away from the cameras. As a result, they perform poorly in acquiring the close-up views of the targets and their activities. HD-CCTV/Megapixel IP cameras have recently been introduced to overcome this resolution issue. Similar to traditional CCTV cameras, these fixed/static HD-CCTV/Megapixel IP cameras are placed to constantly focus at specific locations in the environment. A relatively large network of such cameras has to be installed in order to observe the targets in any region of the environment at high-resolution, which is impractical in terms of equipment, installation, and maintenance costs.

The use of active PTZ (Pan/Tilt/Zoom) cameras is becoming an increasingly popular alternative to that of fixed/static cameras for surveillance because the active cameras are endowed with pan-tilt-zoom capabilities that can be exploited to focus on and observe the targets at high image/video resolution. Hence, fewer active cameras need to be deployed to be able to capture high-resolution images/videos of the targets in any region of the environment. Most of the activities of interests are sporadic in nature and are scattered across the surveillance environment. Therefore, the active cameras can be steered and zoomed to focus on these activities at a high-resolution. Manual control of these cameras in the above applications becomes difficult, especially when the number of targets and cameras increases. Figure 1.1 shows the images of Axis 214 PTZ cameras that are widely used in research and commercial purposes.

Figure 1.1: Axis 214 PTZ cameras.

In order to achieve effective real-time surveillance, an efficient collaborative mechanism is required to control and coordinate these cameras' actions, which is the main focus of this thesis. Figure 1.2 illustrates an example scenario to depict the grand vision of this research with 5 PTZ cameras.

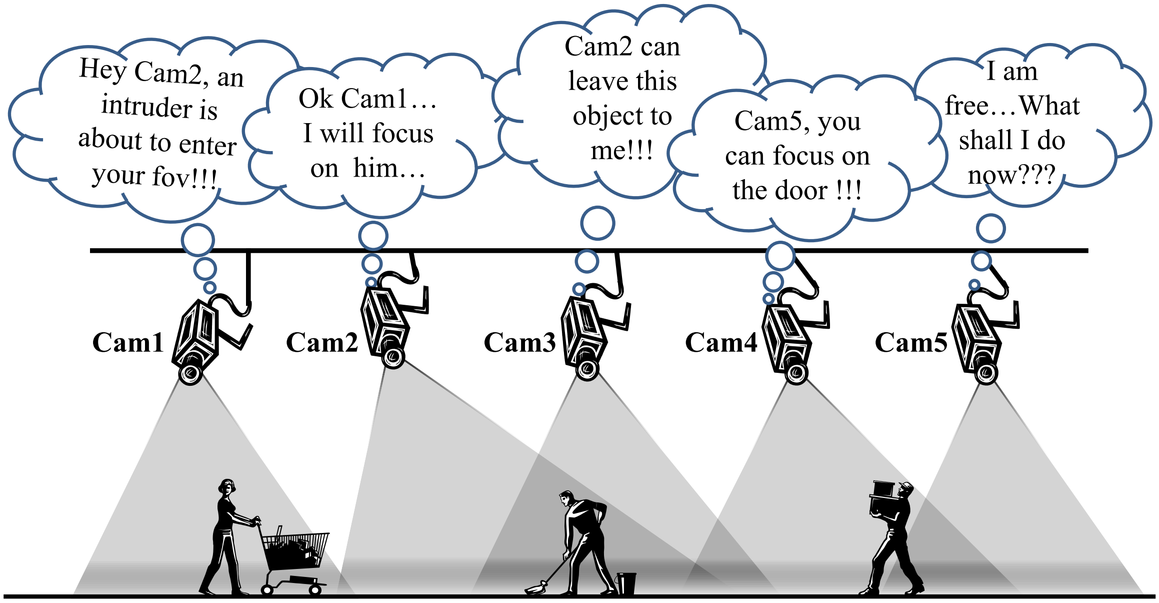

Hey Cam2, an intruder is about to enter your fov!!!
Ok Cam1... I will focus on him...
Cam2 can leave this object to me!!!
Cam5, you can focus on the door !!!
I am free...What shall I do now???
Cam1
Cam2
Cam3
Cam4
Cam5

Figure 1.2: Multiple active camera control and coordination.

1.2 Objective

This thesis aims to address the following central problem in surveillance:

> How can a network of active cameras be coordinated to monitor a set of moving targets with a guaranteed image resolution?

Monitoring a set of targets with a guaranteed resolution is an important surveillance task. Coordinating active cameras in order to observe these targets with a guaranteed image resolution is challenging and non-trivial. This is due to the following practical issues in designing a coordination framework for active cameras in surveillance:

- **Multiple sources of uncertainties:** The surveillance environment is fraught with multiple sources of uncertainties such as targets' stochastic motion, unknown targets' locations, noisy camera observations, etc. These uncertainties in the surveillance environment make it difficult for the active cameras to know where to observe in order to keep the targets within their fields of view (fov). Consequently, they may also lose track of the observed targets.
- **Camera - Target ratio:** In practice, the number of targets to be observed is usually much greater than the number of available active cameras. When the number of targets[1] increases, the camera coordination framework, if poorly designed, tends to incur exponentially increasing computational time, which degrades the performance of the entire surveillance system.
- **Trade-off between maximizing the expected number of targets and the resolution of these observed targets:** Increasing the resolution of observing some targets through panning, tilting, or zooming may result in the loss of other targets being tracked and increasing the number of targets to be observed by decreasing the zoom level, may result in the decrease of the resolution of these targets. Therefore, it is necessary to address this trade-off in the underlying camera coordination framework.
- **Scalability:** The camera coordination framework should be scalable with an increasing number of targets and cameras. The computational time required for calculating optimal control decisions for the cameras should be made in polynomial time for increasing number of targets and cameras. In addition, coordination framework should be scalable to the communication overhead among the cameras.
- **Real-time:** The control decisions for these cameras should be made proactively in real-time.
- **Occlusions:** Many real-world surveillance environments contain obstacles like pillars,

[1] In the rest of the thesis, the term ”number of targets” will be referred to ”number of targets to be observed” in the active cameras.

walls and barriers that occlude the fov of some or perhaps even all of the cameras. This can also be due to privacy issues in monitoring certain regions of the environment. Therefore, it is highly impractical for the cameras to persistently track the observed targets in such environments. The regions where the targets cannot be observed by any of the cameras due to obstacles or privacy issue are referred as *blind regions* and the surveillance environment with *blind regions* is called as *partially observable* environments (see Figure 1.3). Hence when the targets are in the *blind regions*, the camera coordination framework has no information about these targets which causes performance degradation of the surveillance system.

- **Fairness property:** Fairness is a vital property in active camera surveillance where the active cameras are coordinated to observe the targets in the surveillance environment such that no target is "starved" of observation by the cameras for a long time. When there is no fairness in coordinating active cameras, then one or more targets may not be observed (starved) for long duration which may lead to a loophole in surveillance, i.e., the behavior of those targets will neither be monitored nor recorded in high-resolution. Hence it is necessary to incorporate fairness property in the coordination framework.

Therefore, the problem of controlling and coordinating a network of active cameras in order to monitor a set of targets is challenging and needs significant research attention.

1.3 Contributions

1.3.1 A Decision-Theoretic Formulation of Camera Coordination Framework for Surveillance

A novel decision-theoretic multi-agent formulation has been proposed for controlling and coordinating multiple active cameras in surveillance [Natarajan, 2012a; Natarajan, 2012b]. Decision-theoretic approach provides formal and principled frameworks to coordinate the planning of active cameras' control decisions under stochastic and partially observable environments (e.g., uncertainty in targets motion and locations) in achieving the desired surveillance objective/task. It models the surveillance task as a stochastic optimization problem in which optimal actions of the cameras are determined such that the utility of the surveillance is increased. The utility of the surveillance can be modeled as formal objective functions, such that the active cameras are coordinated to achieve this high-level surveillance goal. The first goal of the surveillance system is to maximize the number of targets observed with guaranteed image resolution. We refer to this surveillance task as **maximizing observations of multiple targets (MOMT)** problem. That is, a network of active cameras coordinating to obtain images/videos of the moving targets in the surveillance environment with guaranteed image resolution. A drawback of this surveillance goal is that one or more targets may not be observed by the active cameras for long duration as the cameras are coordinated to focus or observe locations where there are more number of targets in the environment. To overcome this limitation, we propose fairness in observation of targets as our second goal of surveillance. That is, no target is "starved" of observation by active cameras for long duration. We refer to this problem as **fairness in observation of multiple targets (FOMT)** problem. The proposed decision-theoretic formulation exploits the inherent properties and structures that are present in our surveillance problems, in order to scale the framework for increasing number of targets and cameras. That is, the assumption that the motion of each target is conditionally independent of other targets and cameras in the environment given the current position, direction and speed of that target. Specifically, we propose the following novel decision-theoretic frameworks to control and coordinate multiple active cameras: (a) Markov Decision Process (MDP) framework to solve the MOMT problem in *fully observable* surveillance environment, (b) Markov Decision Process (MDP) framework to solve the FOMT problem in *fully observable* surveillance environment, and (c) Partially Observable Markov Decision Process (POMDP) to solve MOMT problem in *partially observable* surveillance environment.

1.3.2 Markov Decision Process (MDP) Framework for Coordinating Cameras

A novel Markov Decision Process (MDP) framework has been proposed to control active cameras in a *fully observable* surveillance environment, i.e., the location, direction and speed of the moving targets are estimated from a set of wide-view static cameras that are calibrated site-wide. In this environment, the targets are assumed to be visible to the static cameras at every instance of time and based on the observations from the static cameras, the proposed MDP framework directs the active cameras to observe the targets in high-resolution. In order to direct the active cameras to the predicted locations of the target, greedy solution (i.e., one step look-ahead of target's motion) has been proposed to solve the underlying MDP. Specifically, the MDP framework resolves some of the above mentioned issues (Section 1.2) in the following ways: (a) the motion of the targets are modeled probabilistically; (b) the non-trivial trade-off between maximizing the expected number of targets and the resolution of these observed targets has been addressed by controlling the active cameras to maximize the number of targets by guaranteeing the predefined image/video resolution; (c) the scalability in number of targets has been improved by exploiting the properties that are present in our surveillance problem; and (d) in order to compute optimal control decisions for cameras in real-time, we pre-compute the solutions off-line and do a look-up operation on our stored solutions during the surveillance. We have also shown that the finite horizon planning (i.e., more than one step look-ahead of uncertainty in target's motion) solution is equivalent to our greedy solution due to the properties that are present in our surveillance system.

One key problem faced by existing multi-camera multi-target surveillance approaches is that of scalability with increasing number of targets. We exploit the structure and properties of our MOMT problem to improve the scalability in number of targets. In order to improve the scalability in number of cameras, we use the concept of Coordination

Graph (CG) [Guestrin *et al.*, 2002] over the active camera network and solve the underlying coordination problem approximately using max-plus algorithm [Vlassis *et al.*, 2004; Kok and Vlassis, 2006]. As shown in simulation, our MDP framework can achieve high-quality surveillance of up to 50 cameras and 60 targets in real-time.

In the above mentioned MDP frameworks, the active cameras are controlled in order to maximize the number of targets observed with guaranteed resolution. A drawback of this task is the lack of fairness in the observation of targets. Therefore, we formally realize a popular fairness metric in resource allocation problems known as max-min fairness, for achieving fairness in active camera surveillance. We extend our MDP framework to optimize this max-min metric, such that no target is "starved" of observation by active cameras for long time.

1.3.3 Partially Observable Markov Decision Process (POMDP) Framework for Coordinating Cameras

A novel Partially Observable Markov Decision Process (POMDP) framework has been proposed to control active cameras in a *partially observable* surveillance environment, i.e., the case where we do not have static cameras that can observe the entire surveillance environment at a low-resolution. Hence, the targets' information are observed only through the active cameras. In such *partially observable* environment, the targets may not be continuously observed in any of the active cameras due to *blind regions* in the surveillance environment. This setup is more realistic because, many real world environments (like airports, railway and subway stations, schools and university campuses, etc.) have occlusions due to physical structures like walls and pillars, and also restricted regions where the cameras cannot be placed. Figure 1.3 shows an example of overhead view of a *partially observable* surveillance setup with 6 active cameras and the occlusion caused by pillars and non-overlapping active cameras.

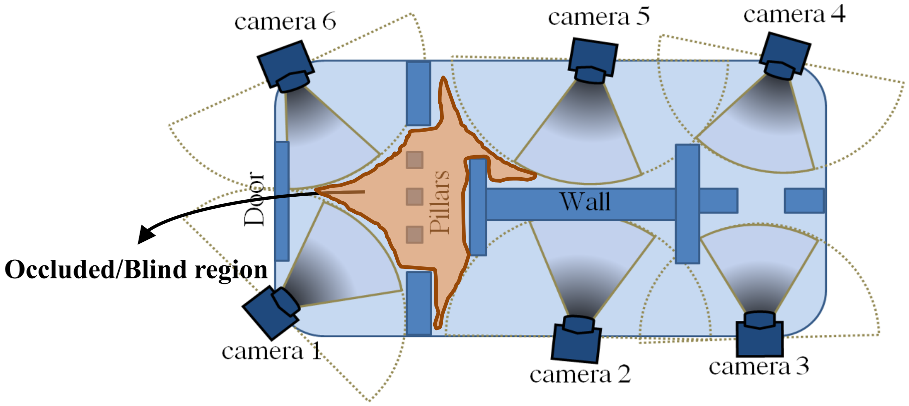


Figure 1.3: An example of *partially observable* surveillance setting.

This framework resolves the above surveillance issues in the following ways: (a) we model the belief over the targets' states (i.e., locations, directions and speeds) and update the belief using the Bayesian paradigm based on the probabilistic models for targets' motion and active cameras' observations; (b) the actions of the active cameras are coordinated to simultaneously improve the belief over the targets states and maximize the expected number of targets observed with a guaranteed pre-defined resolution; (c) the targets' motion uncertainty is modeled by a probabilistic motion model; (d) the noisy camera observation is modeled by having a probabilistic observation model; (e) the non-trivial trade-off between maximizing number of targets and the image resolution of observing these targets is addressed by coordinating the cameras' action such that the expected number of targets is maximized while maintaining a guaranteed image/video resolution; (f) the scalability in number of targets is improved by exploiting the properties in our MOMT problem and (g) the optimal cameras' actions are computed in real-time by using sparse data structures to store and manipulate the probabilities. In this framework, we proposed a greedy solution for MOMT problem in *partially observable* surveillance environment that is scalable in number of targets. As shown in simulations, our POMDP framework can scale up to 20 targets in real-time.

The real camera experiments of our MDP and POMDP frameworks, show the feasibility of our decision-theoretic approach in real world surveillance. The related works on existing camera coordination approaches in the literature related to our contribution in Section 1.3.1 are discussed in Section 2.1. And the related works pertaining to our contributions in Section 1.3.2 and 1.3.3 are discussed in Section 2.2.

1.3.4 Summary of Contributions

To summarize, this thesis has the following novel contributions:

1. We have proposed a novel decision-theoretic approach to control and coordinate multiple active cameras in surveillance system [Natarajan, 2012a; Natarajan, 2012b]. Specifically we have formulated (a) novel surveillance tasks of maximizing observation of multiple targets with guaranteed image resolution both in *fully observable* and *partially observable* surveillance environments; and (b) a novel task of achieving fairness in observation of targets in active camera surveillance under *fully observable* surveillance environment. (see Chapter 2)

2. We have proposed a novel Markov Decision Process (MDP) framework [Natarajan *et al.*, 2012a] to control and coordinate multiple active cameras in a *fully observable* surveillance environment. In this work, we have accounted for targets' motion uncertainty, trade-off between number of targets and the resolution of observing them, and the scalability in number of targets. We have derived a greedy solution for MOMT problem that is scalable with an increasing number of targets. We have also shown in theory, that the finite horizon planning solution for MOMT problem is equivalent to our greedy solution. (see Chapter 3)

3. We have proposed a scalable MDP framework to coordinate active cameras for a large number of cameras. This framework is based on the concepts of coordination

graph and the max-plus algorithm. We decompose the centralized coordination problem involving all cameras into many local coordination problems involving only few cameras. The max-plus algorithm has been used to solve the coordination problem through message passing between cameras. (see Chapter 4)

4. We have proposed a novel MDP framework to control and coordinate active cameras in *fully observable* surveillance environment such that, all the targets are observed by active cameras in a fair manner. To achieve this, we have used max-min fairness metric for observing targets in our coordination framework. The scalability in number of targets is achieved by exploiting the structures and properties in the FOMT problem. We have derived a greedy solution for our FOMT problem in *fully observable* environment that is scalable in the number of targets. (see Chapter 5)

5. A novel Partially Observable Markov Decision Process (POMDP) framework [Natarajan *et al.*, 2012b] has been proposed to control active cameras in a *partially observable* surveillance environment. This framework addresses a key challenge of coordinating multiple active cameras in surveillance when the targets cannot be observed at every time step due to occlusions in the environment. In this framework, we account for the uncertainties due to noisy camera observation, targets' motion, location and its direction. We also eliminate the strong assumption of having many static wide-view cameras that can always observe the targets in the surveillance environment. We have derived a greedy solution for our MOMT problem in *partially observable* environment that is scalable in number of targets. (see Chapter 6)

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 reviews the existing camera coordination approaches in the literature (Section 2.1) and previous works related to our MOMT and

FOMT problems (Section 2.2). In Chapter 3, we formalize the MDP framework for MOMT problem in *fully observable* surveillance environment that can scale for increasing number of targets. In Chapter 4, we extend the MDP framework in Chapter 3 using the coordination graph and max-plus algorithm to improve the scalability in number of cameras. Chapter 5 discusses the MDP framework for FOMT problem in *fully observable* surveillance environment. In Chapter 6, we discuss the POMDP framework for MOMT problem in *partially observable* surveillance environment. Finally in Chapter 7, we summarize the thesis and discuss some of the future works.

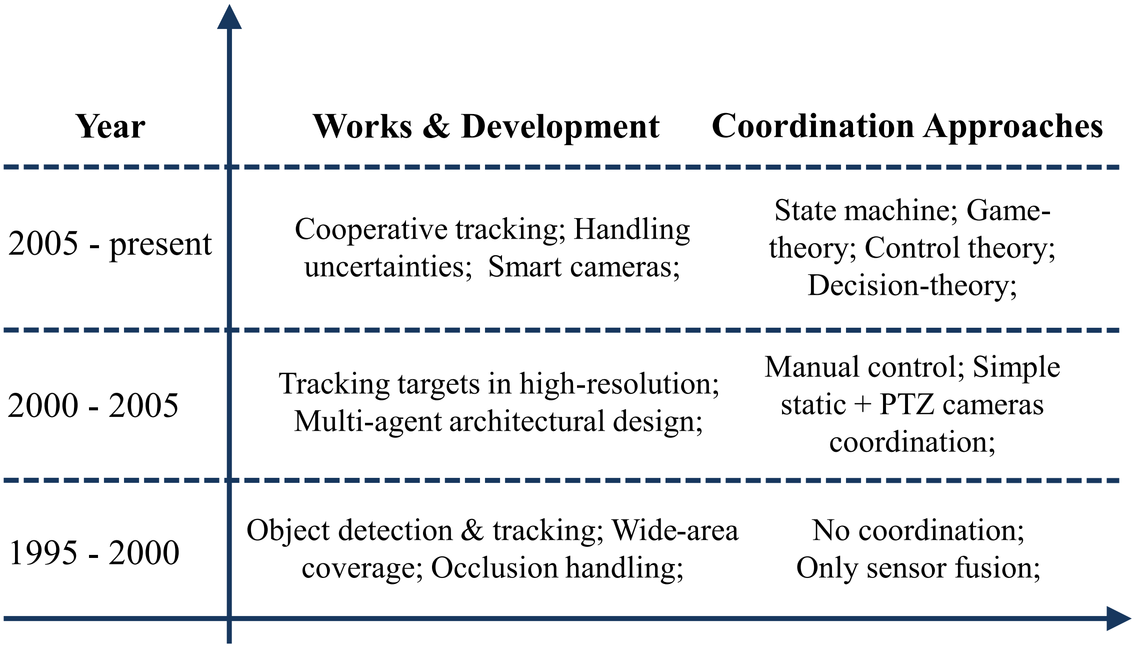
Chapter 2

Literature Review

In this chapter, we review the existing multi-camera coordination methodologies by organizing them into (a) camera coordination approaches and (b) the work related to MOMT and FOMT surveillance problems. Firstly, in Section 2.1 we briefly explore the existing camera control and coordination approaches, and provide their evolution in the past decades. Secondly, in Section 2.2 we compare and contrast the existing methods that are related to our decision-theoretic solutions for MOMT and FOMT surveillance problems.

2.1 Camera Coordination Approaches

In this section, we provide a brief overview on the related work on multi-camera control and coordination. We present the time-line of these coordination approaches along with their applications in surveillance as shown in Figure 2.1. It can be seen from Figure 2.1, the initial work on multiple cameras were started in the 1990s to solve computer vision problems like object detection, recognition, etc. Researchers started using multi-modal cameras to increase the efficiency of their vision algorithms and no coordination between the cameras were considered. For example, [Nandhakumar and Aggarwal, 1988] has used visual images and thermal images for object detection and classification. In [Nandhakumar, 1992], a robust method for object recognition has been proposed in multi-sensory computer vision systems which overcomes the limitations in measuring the thermal features of the object.



Year	Works & Development	Coordination Approaches
2005 - present	Cooperative tracking; Handling uncertainties; Smart cameras;	State machine; Game-theory; Control theory; Decision-theory;
2000 - 2005	Tracking targets in high-resolution; Multi-agent architectural design;	Manual control; Simple static + PTZ cameras coordination;
1995 - 2000	Object detection & tracking; Wide-area coverage; Occlusion handling;	No coordination; Only sensor fusion;

Figure 2.1: Timeline of research and development in multi-camera control and coordination in surveillance.

Later in 2000s, due to the developments in embedded systems, the cost of cameras and other sensors dropped significantly. It was realized that the surveillance coverage area could be increased by adding additional cameras. Object detection and tracking across wide-area was the main focus of using multiple cameras. There was no coordination among the cameras, and only fusion of data from multiple cameras were considered. Hence multi-camera fusion played a major role in multi-camera systems. Multiple cameras were used for extending the coverage area and then the tracking data were combined from multiple cameras at a central server. For example, in [Yuan *et al.*, 2003; Petrushin *et al.*, 2006; Kerhet *et al.*, 2007; Alahi *et al.*, 2008; Fiore *et al.*, 2008], the authors have used multiple cameras to increase the coverage area for object detection and to distribute the processing power across the camera nodes. In [Fukuda *et al.*, 2000; Mittal and Davis, 2001; Scheunert *et al.*, 2004; Hayet *et al.*, 2005; Cavallaro, 2005; Paek *et al.*, 2007; Muoz-Salinas *et al.*, 2009a; Muoz-Salinas *et al.*, 2009b; Krumm *et al.*, 2000; Chang and Gong, 2001; Hu *et al.*, 2006; Guler *et al.*, 2003; Muller and Anido, 2004; Ko and Berry, 2005; Prati *et al.*, 2005; Arsic *et al.*, 2008], the single camera tracking algorithms have been replicated in multiple cameras and the tracking data were fused at the central server. The main motivation of these methods was to increase the tracking area geographically and handle occlusion.

Until early 2000s, most work on object detection/tracking were done with static wide-view cameras, where the cameras are fixed and hence they cannot capture the targets in high-resolution, especially when the targets are far away from the cameras. After the development of PTZ cameras, capturing high-resolution images of targets and their activities became popular. Initially, these PTZ cameras were controlled manually by the user, using the joy-stick and other hardware devices. Based on the videos from static cameras, the user in the security office controls the PTZ cameras to focus on regions of interests in high-resolution [Collins *et al.*, 2001]. For example, [Liu *et al.*, 2002; Lu and Payandeh, 2008] developed a surveillance system know as FLYSPEC system that converts the user selected region of interest in the wide-view cameras into the corresponding PTZ control signals and sends it to the PTZ cameras.

The above setup was improved by controlling the PTZ cameras to automatically gaze the targets that are observed in the static cameras [Micheloni *et al.*, 2005; Ahmedali and Clark, 2006] and omni-directional cameras [Khoshabeh *et al.*, 2007]. These static wide-view cameras and omni-directional cameras are calibrated with the PTZ cameras on a common ground plane coordinates. Based on this simple setup (i.e., coupling static wide-view cameras and PTZ cameras), there has been many work that have been explored in multi-camera control and coordination (see Table 2.1). They are based on the following approaches:

1. Control Theoretic Approach
2. Game Theoretic Approach
3. Heuristic Approach
4. Multi-Agent System Approach

2.1.1 Control Theoretic Approach

Control theory deals with the behavior of dynamic systems that consists of system, sensor measurements and a controller. The control theory induce a feedback mechanism in which the system is controlled based on its measured or observed state. The desired output of the system is called as *reference*. In a typical feedback control mechanism, the controller computes the error between the measured output and the reference, and selects the optimal control signals to minimize the error. Some of the feedback control mechanisms are P (proportional), PI (proportional, integral), PID (proportional, integral and derivative) controllers, etc.

For example, [Wang, 2003] have used proportional feedback controller to control the PTZ camera to center the human face through experiential sampling. Since a proportional controller has a slow response time and is sensitive to noise, [Wang *et al.*, 2011] have adopted the PID feedback controller for centering and tracking the human face to obtain its best-view. [Singh *et al.*, 2008] developed an active and cooperative multi-camera framework that uses Model Predictive Controller as feedback mechanism that allows the cameras to react based on past and future events.

2.1.2 Game Theoretic Approach

Game theory is a branch of decision theory that deals with the interdependent decisions while optimizing the desired goal. It provides a theoretical basis for multi-agent system.

Table 2.1: List of representative works in multi-camera control and coordination.

The work	Coordination Approaches	Cameras	Highlights
[Singh *et al.*, 2008]	Control Theoretic	PTZ	Model Predictive Controller for coopetitive tracking.
[Wang, 2003]	Control Theoretic	PTZ	Proportional Controller for face tracking.
[Wang *et al.*, 2011]	Control Theoretic	PTZ	PID Controller for best-view selection.
[Song *et al.*, 2008a]	Game Theoretic	PTZ	Focus one target in high-resolution and track other targets in wide-view.
[Jones and Mitter, 2006]	Game Theoretic	generic sensors	Players as sensors and action of players as sensor tasks.
[Li and Bhanu, 2011]	Game Theoretic	PTZ	Focus one target in high-resolution and track other targets in wide-view.
[Ding *et al.*, 2012a]	Game Theoretic	PTZ	Prioritize tasks for cooperative tracking of targets.
[Ding *et al.*, 2012b]	Game Theoretic	PTZ	Maximize tracking accuracy, best-shot and image resolution.
[Qureshi and Terzopoulos, 2008]	State Machine	static+PTZ	Scheduling active cameras; ContractNet protocol for group formation.
[Qureshi, 2010]	State Machine	static+PTZ	Collaborative sensing tasks; Negotiation decision making.
[Starzyk and Qureshi, 2011b]	State Machine	static+PTZ	Behavior based camera controller.
[Starzyk and Qureshi, 2011a]	State Machine	static+PTZ	Learning camera control strategies based on past actions.
[Ilie and Welch, 2011]	Constrained Optimization	PTZ	Stochastic performance metric and constrained optimization for target tracking.
[Piciarelli *et al.*, 2009]	EM algorithm	static+PTZ	Camera control for event/activity; Expectation Maximization algorithm to choose PTZ parameters.
[Dieber *et al.*, 2011]	EM algorithm	static+PTZ	Coverage optimization and resource allocation.
[Sommerlade and Reid, 2010]	Probabilistic measure	static+PTZ	Scene exploration and tracking.
[Krahnstoever *et al.*, 2008]	Probabilistic measure	static+PTZ	Asynchronous optimization and combinatorial search.
[Collins *et al.*, 2001]	User Control	static+PTZ	GUI for controlling the sensors; 2D and 3D output visualization.
[Fleck *et al.*, 2006]	User Control	static+PTZ	Master-slave configuration; GUI for controlling the cameras.
[Hodge and Kamel, 2003]	Agent-based	static	Surface coverage and feature extraction of targets.
[Castanedo *et al.*, 2006]	Agent-based	static	Increase accuracy of tasks and accomplish tasks that cannot be done by single camera.
[Garcia *et al.*, 2005]	Agent-based	static	Solve a surveillance tasks; two-roles: camera-role and fusion role; cueing and handoff tasks.
[Matsuyama and Ukita, 2002]	Agent-based	static	Layered architecture with active vision-agents.
[Bustamante *et al.*, 2013]	Agent-based	PTZ	Integrated fusion architecture for collaborative camera control.

A *Game* in game theoretic model is a mathematical object that consists of set of players, moves for the players and the utility for the combination of the moves of the player.s Solving the game is to identify the sequence of moves that the players should use. A sequence of moves is called a strategy, so an optimal strategy is a sequence of moves that results in the best outcome. Each player in the game, makes their own best move taking into the consideration of the best moves of every other players. This concept is known as Nash equilibrium. Game theory is used in many surveillance problems in order to solve cooperative and competitive target tracking, where the cameras are modeled as players and utility of the game is to track the targets. The optimal strategy of the game is to choose actions of the active cameras such that the surveillance goal is achieved.

[Song *et al.*, 2008b] have demonstrated the guidelines for adopting game theoretic framework for integrated sensing and analysis in a distributed multi-camera network. They try to maximize the area coverage in each of the cameras using game-theory based distributed optimization and consensus algorithms. [Li and Bhanu, 2011] had proposed a game theoretic framework for camera assignment and hand-off in video network. [Jones and Mitter, 2006] adopted game theoretic approach for sensor coordination, to determine optimal computation and communication necessary for group coordination among the sensors. [Song *et al.*, 2008a] presented a decentralized camera control algorithms for accurate and efficient target acquisition using game theory approach which was then evaluated by [Zongjie and Bhattacharya, 2011] for different camera configurations. [Soto *et al.*, 2009] proposed a game theoretic control of PTZ cameras to track targets at high-resolution and also observe the other parts of the environment in acceptable resolution. [Ding *et al.*, 2012b] have proposed an integrated analysis and control framework for PTZ camera network in order to maximize various scene understanding objectives like tracking accuracy, best-shot, and image resolution. [Ding *et al.*, 2012a] proposed a method to prioritize tasks for a distributed camera network to co-operatively track and acquire high-resolution images of the targets. [Morye *et al.*, 2013] proposed a camera control mechanism to obtain opportunistic high-resolution facial images through distributed constrained optimization techniques.

2.1.3 Heuristic Approach

The camera control framework which does not fall into above frameworks are classified as heuristic frameworks. These frameworks are based on ad-hoc optimization techniques, state machine approaches, probabilistic approach, etc. These works are tailored specifically to address their own surveillance task and is not obvious to adopt for other surveillance tasks.

[Sommerlade and Reid, 2010] has given a probabilistic framework for controlling multiple active cameras based on mutual information of the scene. [Ilie and Welch, 2011] proposed a stochastic optimization method for controlling the active cameras for tracking targets. They explore on the optimal camera configurations over time, predicted trajectories and the underlying image processing algorithms. [Piciarelli *et al.*, 2009] has proposed an optimization technique to determine the optimal pan/tilt/zoom parameters of the active cameras to investigate a particular event based on activity map. [Dieber *et al.*, 2011] has proposed an expectation-maximization (EM) algorithm for optimizing the coverage for a given set of regions-of-interests and allocate resources efficiently in PTZ camera networks. [Krahnstoever *et al.*, 2008] has presented a system that controls a set of PTZ cameras to acquire the close-up video of the people in the surveillance environment. They use set of static cameras to localize the people in the environment and assign each of the pedestrian to freely available PTZ cameras. [Micheloni *et al.*, 2010] discussed the video analysis in PTZ camera networks and mainly focus on the signal processing techniques for tracking and localizing targets using cooperative PTZ cameras.

[Qureshi and Terzopoulos, 2008] has presented a distributed camera control strategy for a smart camera network that consists of uncalibrated static and active cameras. The authors had presented the preliminary work of scheduling smart cameras in [Qureshi and Terzopoulos, 2005]. In these works, the camera node behavior has been modeled using a finite state machine. The ContractNet protocol has been used to model the group formation among the camera nodes in distributed control strategy. When the user selects a target to track, the sensors coordinate among themselves to form a dynamic group to fulfill the task. [Qureshi, 2010] has given a negotiation protocol for collaborative sensing tasks of PTZ cameras in a distributed manner. [Qureshi and Terzopoulos, 2009] has proposed a classical planning approach to control active cameras in order to get high-resolution video of pedestrians supported by static wide-view cameras. This classical planning approach predicts the future trajectories of targets and assigns PTZ cameras to track the targets, such that there are few camera assignments and hand-offs in future.

[Starzyk and Qureshi, 2011b] try to learn the control strategy of the PTZ cameras based on the previous control actions and in [Starzyk and Qureshi, 2011a] they proposed a behavior based camera controller in a distributed surveillance system to handle multiple observation tasks simultaneously.

2.1.4 Multi-Agent System Approach

Multi-agent framework consists of multiple agents that perceive and act upon the environment. These agents interact among each other to solve a problem which cannot be solved by individual agents. The agents in Multi-Agent System (MAS) refers to many entities including software components, humans, human teams, robots, sensors, etc. In surveillance applications, individual camera or set of cameras refers to an *agent*. MAS provides a control framework for controlling distributed autonomous cameras in many wide-area surveillance problems. The BDI mode (belief, desire, intention) of MAS assists in incorporating high-level application goals of the surveillance system into the agents (cameras). The belief represents the agents' information about the state of the environment, the desire is the motivational state of the agents and intentions are the current actions of the agents.

As the starting point, the previous work in multi-camera systems provided a high-level agent-based architectures to model the control and coordination between each camera agent. For example, [Matsuyama and Ukita, 2002] presented a three-layered dynamic interaction architecture for cooperative multi-target tracking system that consists of group of Active Vision Agents (AVA) distributed across the environment. [Bramberger *et al.*, 2005] integrated multi-camera tracking into an agent-based dynamic task allocation system for traffic surveillance. They have used distributed embedded system with limited resources that consists of *smart camera*[1] nodes. [Garcia *et al.*, 2005] have presented an agent based architecture to solve the camera coordination problem for surveillance tasks. In this approach, each agent is an autonomous software module that controls the operation of a camera based on the video frames and the messages from its neighboring agents. [Castanedo *et al.*, 2006] presented an architecture for Cooperative Sensor Agents (CSA) and a coalition protocol for coordination of these CSA. In this work, each CSA is an autonomous agent and collaborate with other CSAs to achieve better performance in completing a task. [Bustamante *et al.*, 2013] proposed a high-level MAS architecture to coordinate multiple active cameras based on the users' inputs. They have integrated fusion architecture to combine data from multiple cameras and control them to achieve the users' desired goals. It can be seen that, the authors have provided agent-based architectures for generic sensors and smart cameras. They have not explored in detail on control and coordination of active cameras which is the focus of this thesis.

[1] The smart cameras consists of hardware and software framework with inbuilt dedicated Digital Signal Processing and Network processor. The smart cameras are embedded with video-sensing, high-level video analysis, compression and communication capabilities (see [Rinner and Wolf, 2008]).

2.1.5 Summary of Camera Coordination Approaches

From the survey, we can see that researchers have adopted different approaches to control and coordinate multiple active cameras like control theory, game theory, state machine, MAS approach, probabilistic approach, and many other ad-hoc approaches as well. In this thesis, we propose a novel decision-theoretic approach to control and coordinate multiple active cameras in surveillance. The nature of the surveillance problem (i.e., choosing optimal actions of active cameras in presence of uncertainties) makes decision-theoretic approach an appropriate choice to control and coordinate active cameras in stochastic surveillance environment. [Spaan and Lima, 2009] have proposed a decision-theoretic approach to select cameras in the surveillance system. Their work is purely based on selecting one or more static cameras which is different from our work on controlling active cameras to accomplish a desired surveillance task. Our decision-theoretic approach offers some of the following advantages in surveillance:

- Decision-theoretic approach models the interaction between active camera network and the surveillance environment effectively. Specifically, it models the surveillance task as a stochastic optimization problem in which optimal actions of the cameras are determined such that the utility of the surveillance is increased.
- It provides formal, principled and rich mathematical models like Markov Decision Process (MDP), Partially Observable Markov Decision Process (POMDP), etc. for planning optimal control actions for cameras in presence of uncertainties like targets' motion and location, noisy camera observations, occlusions, etc.
- Multiple high-level surveillance goals can be defined formally as mathematical objective functions.

When decision-theoretic models are poorly designed or used naively for a surveillance problem, the state space explodes and hence computing optimal actions for these cameras becomes intractable. For example, [Spaan and Lima, 2009] has serious limitation in terms of scalability in number of targets and cameras. Whereas in our thesis, we exploit the structures and properties of the underlying surveillance problem to improve the scalability both in number of targets and cameras.

2.2 Related Works on MOMT and FOMT Problems

In this section, we compare and contrast previous work in active camera network that are related to our proposed methodologies: (a) MDP for MOMT (Chapter 3), (b) MDP for FOMT (Chapter 5) and (c) POMDP for MOMT (Chapter 6). Table 2.2 provides the comparison with the existing works in key terms of:

- **Camera - Target ratio:** The ratio of number n of active cameras to number m of targets that is used. This ratio is further classified into $n \gg m$, $n \ll m$ and $n = m$. The camera - target ratio plays a crucial role in controlling the active cameras because, camera control becomes interesting and more challenging when $n \ll m$, i.e., more number of targets should be monitored by few active cameras.
- **Primary criterion:** The primary criterion reflects the goal of the surveillance system in which the active cameras are controlled. This is further classified into (a) maximizing the number of observed targets with certain guaranteed resolution, (b) Focusing and tracking individual target in high-resolution and (c) observing the targets in a fair manner so that no target is "starved" of observation by active cameras for long duration. In this thesis, the goal of the surveillance is to maximize the number of targets observed with guaranteed image resolution in Chapter 3, 4 and 6, whereas in Chapter 5 our surveillance goal is to observe all the targets in active cameras in a fair manner.
- **Uncertainty:** Whether the uncertainty of targets' motion and location are accounted in the camera coordination framework. The targets' motion is stochastic in nature and hence needs to be predicted and subsequently exploited for coordinating the cameras in a typical surveillance system. Similarly, the locations determined by the cameras are often

not accurate due their poor resolution and the underlying calibration model. In *partially observable* surveillance environment, the locations of targets are highly uncertain when these targets are in *blind regions*. Hence it is necessary to model and account for the target's location uncertainty in the coordination framework.

- **Surveillance environment:** The surveillance environment in which the camera coordination framework can work is classified into (a) *fully observable* and (b) *partially observable*. In *fully observable* environment, the states of the targets (i.e., location, direction and speeds) are always observed by having many static wide-view cameras or making one or more active cameras to zoom out. Whereas in *partially observable* environment, the cameras can observe only part of the surveillance environment and cannot observe all the targets at every time step due to *blind regions* and privacy issues as stated in Chapter 1.

2.2.1 Related Work on MOMT Problem

In this section, we first discuss the difference between our MDP framework in Chapter 3 and existing camera control approaches. Then we discuss the difference between our MDP framework in Chapter 4 and other related work on large-scale active camera networks. Finally we discuss the difference between our POMDP framework in Chapter 6 and other related work in active camera network.

As shown in Table 2.2, when the camera-target ratio is either $n = m$ [Banerjee *et al.*, 2010] or $n \gg m$ [Banerjee *et al.*, 2010; Costello and Wang, 2005; Krahnstoever *et al.*, 2008; Qureshi and Terzopoulos, 2009; Soto *et al.*, 2009; Sommerlade and Reid, 2010; Huang and Fu, 2011; El-Alfy *et al.*, 2009; Ward and Naish, 2009a; Collins *et al.*, 2001; Hampapur *et al.*, 2003; Starzyk and Qureshi, 2011b; Starzyk and Qureshi, 2011a], the primary criterion is to focus and track one or more targets in a close-up view. In contrast,

Table 2.2: Comparison of previous work related to MOMT and FOMT problems (n: number of active cameras; m: number of targets; **MN**: Maximizing number of targets, **FT**: Focus and Track few targets, **FP**: Fairness Property, **UM**: uncertainty model for targets' motion, **UL**: uncertainty model for targets' location, **FO**: Fully observable environment (static and PTZ cameras), **PO**: Partially observable environment (only PTZ cameras)).

Surveillance/tracking strategy	$n \ll m$	$n \gg m$	$n = m$	**MN**	**FT**	**FP**	**UM**	**UL**	**FO**	**PO**
[Banerjee *et al.*, 2010]			×		×				×	
[Costello and Wang, 2005]		×			×				×	
[Krahnstoever *et al.*, 2008]		×			×				×	
[Qureshi and Terzopoulos, 2009]		×			×				×	
[Soto *et al.*, 2009]		×			×				×	
[Sommerlade and Reid, 2010]		×			×				×	
[Huang and Fu, 2011]		×			×				×	
[El-Alfy *et al.*, 2009]		×			×				×	
[Ward and Naish, 2009a]		×			×				×	
[Collins *et al.*, 2001]		×			×				×	
[Hampapur *et al.*, 2003]		×			×				×	
[Starzyk and Qureshi, 2011b]		×			×				×	
[Starzyk and Qureshi, 2011a]		×			×				×	
MDP for MOMT (Chapter 3)	×			×			×		×	
MDP for FOMT (Chapter 5)	×					×	×		×	
POMDP for MOMT (Chapter 6)	×			×			×	×	×	×

when the camera-target ratio is $n \ll m$, the primary criterion is to maximize the number of observed targets in the environment as we consider in Chapter 3. In either criterion, the targets' motion is inherently non-deterministic. But, none of the previous work have accounted for the motion uncertainty in their optimization framework by a suitable probabilistic model. These works use heuristic approaches to select the best actions for the active cameras. Such approaches are therefore tailored specifically to their own objectives and cannot be modified to achieve other objectives. In contrast, our approach is a general framework in which different surveillance goals can be modeled as formal objective functions. Many previous works ([Krahnstoever *et al.*, 2008; Qureshi and Terzopoulos, 2009; Sommerlade and Reid, 2010; Spaan and Lima, 2009], etc.) face scalability issue in terms of the number of targets to be observed. Whereas in our MDP framework in Chapter 3 [Natarajan *et al.*, 2012a], we have exploited the structure of our surveillance problem in order to make it scalable for increasing number of targets.

In Chapter 4, we address the scalability in number of cameras along with the other issues addressed in Chapter 3. Now, we discuss the works related to our MDP framework in Chapter 4 that is scalable for large number of active cameras. [Bustamante *et al.*, 2012] describes the distributed architecture for coordinated operations of multiple sensors based on centralized fusion and decision making system. [Qureshi and Terzopoulos, 2008] proposed a camera control framework that distributes the processing to each of the cameras and resolves the conflicts by using Constraint Satisfaction Problem in a central server. These works are mainly focused on architecture level of PTZ camera coordination, and more over their coordination framework uses rule-based approach. As mentioned in Section 2.1.2, Game theory based distributed control of active cameras has been explored in [Song *et al.*, 2011c; Song *et al.*, 2011b; Ding *et al.*, 2012a; Ding *et al.*, 2012b]. They formulate the coordination problem as a multi-player game, where each camera is a player and interested in optimizing only its own utility by exchanging their current PTZ settings with all other cameras in the network and try to obtain Nash equilibrium. In contrast, our work in Chapter 4 is based on concepts of Coordination Graph and Max-plus algorithm. Coordination graph is an undirected graph that is used to express the coordination structure of active camera network. In max-plus algorithm, each camera exchanges the messages (which is the utility function) only with their neighboring cameras and improves the approximate global utility function iteratively.

Max-plus algorithm has been explored in image processing and robotics communities. In [Freeman and Liu, 2011; Lan *et al.*, 2006; Duchi *et al.*, 2007], the max-product algorithm (which is analogous to max-plus) has been used to estimate MAP (maximum a posteriori) through belief propagation techniques. Belief propagation operates by passing messages between the nodes in the Markov random fields and are used in image restoration, denoising and super-resolution applications. In [Kok and Vlassis, 2006; Vlassis *et al.*, 2004], max-plus algorithm has been extensively studied and applied to coordinate multiple robots in RoboCup, and also used in optimizing the behavior of traffic lights in urban cities [Medina and Benekohal, 2012; Kuyer *et al.*, 2008]. The message passing strategy and the decomposition of global utility function into sum of local utility functions based on the Coordination Graph structure, makes the max-plus algorithm an appropriate choice to extend our MDP framework in Chapter 3 to improve the scalability in number of cameras in Chapter 4.

The MDP frameworks discussed in Chapter 3 and 4 have constrain of having static cameras to fully observe and track targets in low-resolution. Whereas in Chapter 6, we relax this constrain by extending our MDP framework in Chapter 3 to a POMDP framework. Now we will discuss the difference between our POMDP framework in Chapter 6 and other camera control approaches in the literature. Existing camera control approaches work in a constrained environment where all the targets should be observed in any of the cameras' fov. We term such surveillance environments as *fully observable* environments. In order to achieve this, they use additional static cameras and sensors ([Costello and Wang, 2005; Krahnstoever *et al.*, 2008; Qureshi and Terzopoulos, 2009; Starzyk and Qureshi, 2011b;

Starzyk and Qureshi, 2011a; Hampapur *et al.*, 2003; Ward and Naish, 2009a]), or configure one or more active cameras to zoom-out in a wide-view ([Banerjee *et al.*, 2010; Sommerlade and Reid, 2010; Huang and Fu, 2011; Soto *et al.*, 2009; Song *et al.*, 2008a]) and determine the locations of the targets in the environment. Based on their locations, they estimate the targets' directions and speeds. They use this targets' information to predict their trajectories in order to schedule or control the active cameras to focus on these targets. The major drawback of these approaches are: (a) their camera coordination approaches cannot be deployed in real world surveillance environments that have occlusions due to external barriers. In this case, the camera coordination frameworks are not aware of the targets that are in the *blind region* and hence limits the active cameras' capability; (b) the static cameras cannot always return the accurate locations of the targets as they are low-resolution wide-view cameras. This in turn induces errors in targets directions and speeds, and consequently affects the prediction capability of the active cameras. Whereas in our POMDP framework in Chapter 6, we track the targets in the belief space which is a probability distribution over targets' locations, directions and speeds. We observe the targets' location only in high-resolution active cameras and update the belief of targets constantly using probabilistic targets' motion model and cameras' observation model. Similar to Chapter 3, we exploit the structures and properties in our surveillance problem to improve the scalability in number of targets for our POMDP framework.

2.2.2 Related Work on FOMT Problem

The active cameras in Chapters 3 and 4 are controlled to observe where there is maximum number of targets in the environment. But in Chapter 5, the cameras are controlled to observe the targets in a fair manner such that no target is "starved" of observation by active cameras. In this section, we discuss the previous works in active camera networks that are related to our FOMT problem.

In active camera literature, the cameras are controlled and coordinated either to focus & track one or more targets at desired resolution [Krahnstoever *et al.*, 2008; Qureshi and Terzopoulos, 2009; Sommerlade and Reid, 2010; Song *et al.*, 2011a] or to maximize the number of targets with guaranteed resolution (see Table 2.2). As mentioned previously, the major issue in these methods is that, one or more targets may be observed for long durations, whereas the remaining targets may be observed for short durations or in the worst case, they may not be observed at all. Such bias of the cameras towards few targets would be unfair to the remaining ones. This is due to the following facts: (a) there are more number of targets to be observed than the available active cameras in surveillance environment, (b) targets may pass through the *blind* regions and (c) the active cameras may only partially cover the surveillance environment. [Costello and Wang, 2005] and [Ward and Naish, 2009b] have evaluated different scheduling policies to observe targets in PTZ cameras. They define the order in which the targets to be observed in active cameras based on scheduling policies like random, first-come-first-served, earliest-deadline-first, etc. But they do not consider the fairness property in target observation. To the best of our knowledge, there is no notion of the fairness property that has been used in observation of targets in multi-camera surveillance. We extend our previous work in Chapter 3 to improve scalability in number of targets to achieve fairness in observation of multiple targets.

2.3 Summary

To summarize, our proposed works are different from the existing works in the following ways: (a) we use formal, principled decision-theoretic frameworks (i.e., MDPs and POMDP) to select the optimal actions for active cameras to maximize the expected number of targets observed and to preserve fairness in observation of targets; (b) we have $n \ll m$ which is challenging and address a non-trivial trade-off between number of targets and their resolution to be observed; (c) we account for the uncertainty in target's motion by integrating a probabilistic motion model and the uncertainty of target's locations by integrating a probabilistic observation model into our optimization framework; and (d) we provide a scalable camera coordination framework in terms of number of targets for both *fully observable* and *partially observable* surveillance environments, and scalable in number of cameras for *fully observable* surveillance environment.

Chapter 3

MOMT Problem in Fully Observable Surveillance Environment

3.1 Introduction

This chapter presents a novel principled decision-theoretic approach to control and coordinate the active cameras for **maximizing observations of multiple targets** (**MOMT**) under *fully observable* surveillance environment. That is, the static cameras monitor the entire surveillance environment and track the targets in low-resolution at every time step. Based on this tracking information, the surveillance system controls and coordinates the active cameras in order to maximize the number of targets with guaranteed resolution. This approach is based on the Markov Decision Process (MDP) framework, which allows the surveillance task to be framed formally as a stochastic optimization problem (Sections 3.2 and 3.3). In particular, our MDP-based approach resolves some of the issues mentioned in Chapter 1: (a) the motion of the targets can be modeled probabilistically (Section 3.3.2), and (b) to address the trade-off between the number of targets observed in active cameras and the resolution of observing them, the active cameras' actions are coordinated to maximize the expected number of observed targets while guaranteeing a pre-defined resolution of these observed targets (Section 3.3.3), and (c) the scalability in number of targets has been improved by exploiting the problem structure: as proven analytically (Section 3.4), our MDP-based approach incurs time that is linear in the number of targets to be observed during surveillance. One key problem faced by existing multi-camera multi-target surveillance approaches is that of scalability with increasing number of targets (Chapter 2). As demonstrated empirically through simulations (Section 3.5.2), our MDP-based approach can achieve high-quality surveillance of up to 50 targets in real-time and its surveillance performance degrades gracefully with an increasing number of targets. The real-world experiments (Section 3.5.3) show the practicality of our decision-theoretic approach to control and coordinate cameras in surveillance systems.

3.2 System Overview

The proposed surveillance framework consists of a supervised surveillance environment and an MDP controller. The environment consists of targets, static cameras, and active cameras. The targets are the moving objects (e.g., people, vehicles, robots, etc.) in the surveillance environment whose motions are stochastic in nature. The static cameras are wide-view cameras that can fully observe the surveillance environment at low-resolution. These cameras are assumed to be calibrated and can obtain the 3D location, direction, and speed information of the targets at every instance of time. The active cameras are PTZ (pan/tilt/zoom) cameras that can get high-resolution images of the targets in the environment. The MDP controller models the interaction between the active cameras and the environment, and provides a platform to choose optimal actions for these cameras in order to achieve high-quality surveillance tasks.

Figure 3.1 shows the top view of a representative surveillance environment where the full fov's of the active cameras are shown in dotted lines and the current active fov's are

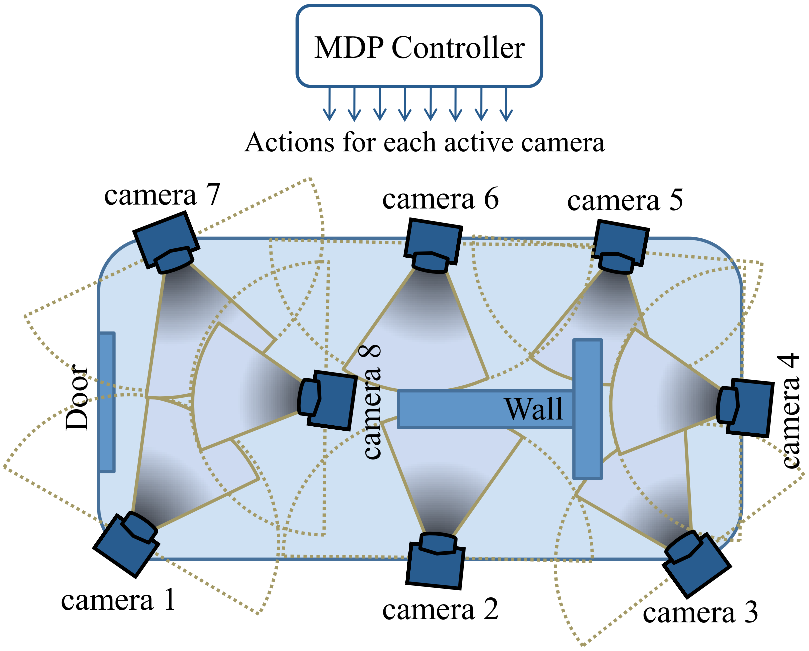


Figure 3.1: System overview of active camera networks in MDP framework. *(The static cameras are assumed to be placed at the ceiling and are not shown in the figure for simplicity.)*

shaded. The static cameras are assumed to be placed at the ceiling and are not shown in the figure for simplicity. The active cameras are placed such that they can observe the complete environment by pan/tilt/zoom operations but cannot observe all locations of the environment simultaneously. This makes the problem more practical and challenging, thus emphasizing the need to control these active cameras. The static cameras determine the location, direction, and speed of targets and pass these information to the MDP controller. Based on these information, the MDP controller computes the optimal actions of active cameras such that the expected utility of the surveillance system is maximized. The MDP framework for controlling active cameras is shown in Figure 3.2. The utility of the surveillance system corresponds to the high-level application goal that can be defined formally using a real-valued objective function, as described in Section 3.3.3.

Formally, the MDP framework is defined as a tuple $(\mathcal{S}, \mathcal{A}, R, T_f)$ consisting of:

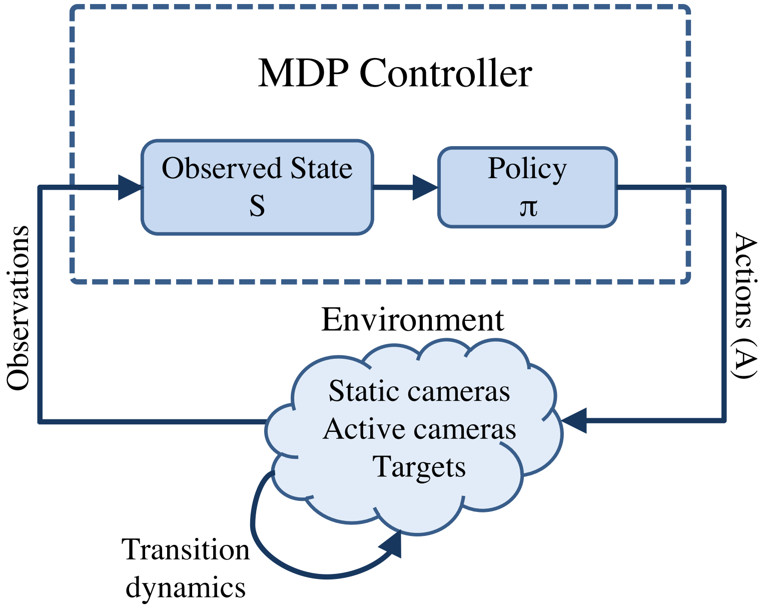


Figure 3.2: MDP framework for controlling active cameras.

- a set $\mathcal{S}$ of discrete states of active cameras and targets in the surveillance environment (Section 3.3.1),
- a set $\mathcal{A}$ of joint actions of active cameras (Section 3.3.1),
- a transition function $T_f : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denoting the probability $P(S'|S, A)$ of switching from the current state $S \in \mathcal{S}$ to the next state $S' \in \mathcal{S}$ using the joint action $A \in \mathcal{A}$ (Section 3.3.2),
- a real-valued reward function $R : S \rightarrow \mathbb{R}$ representing the high-level surveillance goal (Section 3.3.3).

In the MDP framework, the policy function $\pi : \mathcal{S} \rightarrow \mathcal{A}$ maps from each state to a joint action of the cameras. Solving the MDP involves choosing the policy that maximizes the expected reward for any given state. The optimal greedy policy, denoted by π_g^*, maximizing the expected utility of the system in the next time step is given by

$$\pi_g^*(S) = \arg\max_{A \in \mathcal{A}} \sum_{S' \in \mathcal{S}} R(S')\ P(S'|S, A)\ . \tag{3.1}$$

The main challenge in the MDP is managing the state space $\mathcal{S}$ and action space $\mathcal{A}$. This is because the state space grows exponentially in the number of active cameras and targets. Hence, the policy computation time for our surveillance problem is exponential. In practice, the structure of the problem and environment can usually be exploited to reduce the number of states and the time required to compute the optimal policy. We will show in Section 3.4 how the state space can be managed for our surveillance problem, thus allowing the MDP to be solved more efficiently. The following assumptions are made in our surveillance task:

- The targets are oblivious to the cameras, in particular, non-evasive (i.e., they do not try to escape from the cameras' fields of view) and their motion cannot be controlled nor influenced;
- The static cameras are calibrated such that the 3D positioning errors of the targets are minimal. This can be achieved by placing the cameras at high altitude;
- The total number of targets in the environment can be obtained from static cameras.

3.3 Problem Formulation

Given a set of cameras and targets in a surveillance system, the MDP controller determines the optimal actions for these cameras such that the expected utility of the surveillance system is maximized. In this section, we describe how an MDP framework can be applied to a generic active camera surveillance in order to maximize the expected utility of the surveillance system. We enumerate each component of the MDP framework and show how these components can be formulated for a typical surveillance system. In this work, the objective/reward function of the MDP modeling the high-level surveillance goal measures the total number of targets observed by active cameras with a guaranteed resolution. Maximizing the number of observed targets with a guaranteed resolution is a mandatory task in surveillance because we need to obtain the high-resolution images of targets for biometric and forensic tasks like target detection, recognition, etc. In this work, we present a decision-theoretic approach for maximizing the expected number of targets observed by the active cameras. Table 3.1 summarizes the mathematical notations and its descriptions used in this formulation.

Table 3.1: Mathematical notations and its descriptions used in Chapter 3.

Notation	**Description**
n	Number of active cameras.
m	Number of targets to be monitored by n active cameras, such that $n \ll m$.
$\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$	State space of a target represented by a set of tuples of location, direction and speed.
$\mathcal{T}_l$	Set of all possible discretized locations of the target in the surveillance environment.
$\mathcal{T}_d$	Set of all possible discretized directions of target.
$\mathcal{T}_v$	Set of discretized speeds of the target.
t_k=$(t_{l_k}, t_{d_k}, t_{v_k})$	State of a target k is a tuple consisting of its location $t_{l_k} \in \mathcal{T}_l$, direction $t_{d_k} \in \mathcal{T}_d$ and speed $t_{v_k} \in \mathcal{T}_v$.
$T \in \mathcal{T}^m$	Joint state of m targets represented by $T = (t_1, t_2, \ldots, t_m)$.
$\mathcal{C}$	State space of a camera that consists of finite set of discretized poses of camera. Each pose of camera is given by its pan, tilt and zoom value.
$c_i \in \mathcal{C}$	State of camera i, which is given by the discretized pan, tilt and zoom value.
$C \in \mathcal{C}^n$	Joint state of n cameras represented by $C = (c_1, c_2, \ldots, c_n)$.

$fov(c_i) \subset \mathcal{T}_l$	Subset of target locations lying within the field of view (fov) of camera i in its state c_i.
$fov(C) \subset \mathcal{T}_l$	Subset of target locations lying within the joint fov of all cameras in state C, i.e. $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$.
$\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$	State space of MDP which consists set of tuples of joint state of m targets and n active cameras.
$S = (T, C) \in \mathcal{S}$	A state of MDP that consists of joint state of m targets and n active cameras.
a_i	Action of camera i is a PTZ command to move the camera to the specified state.
A	Joint action of all active cameras represented by a tuple $A = (a_1, a_2, \ldots, a_n)$.
$\mathcal{A}$	Set of joint actions of all active cameras.

3.3.1 States and Actions

A state of the MDP comprises the states of active cameras and targets in the surveillance environment. The passive static cameras are first calibrated based on common ground plane coordinates and then used to obtain the targets' approximate 3D location, speed, and the direction information. Let n be the number of active cameras and m be the number of targets in the environment such that $n \ll m$. In this manner, the surveillance problem becomes more challenging and interesting since there are more targets to be monitored by fewer active cameras.

Let the set of possible states of each active camera in the environment be denoted by $\mathcal{C}$ such that each state $c_i \in \mathcal{C}$ corresponds to a discretized pan/tilt/zoom position of camera i. For example, in Figure 3.3(a), the set of possible states of camera i based on discretized pan angles is given by $\mathcal{C} = \{+90\,^{\circ}, +45\,^{\circ}, 0\,^{\circ}, -45\,^{\circ}, -90\,^{\circ}\}$ and the current state c_i is $+45\,^{\circ}$.

Let the state space of a target be represented by a set of tuples of location, direction and speed, and denoted by $\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$ where $\mathcal{T}_l$ denotes a set of all possible locations of

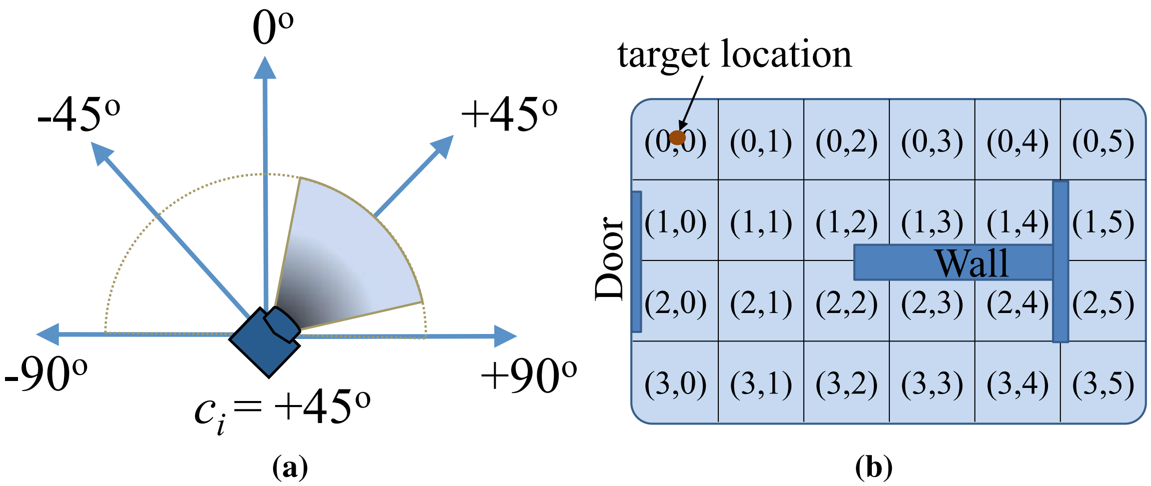


Figure 3.3: (a) Camera states and (b) target locations.

the target in the environment, $\mathcal{T}_d$ denotes a set of all possible discretized directions between all pairs of locations in $\mathcal{T}_l$, and $\mathcal{T}_v$ denotes a set of discretized speeds of the target. The surveillance environment is discretized into grid cells such that the centers of the grid cells represent the possible locations of a target, as shown in Figure 3.3(b). The approximate 3D location of the target observed by static cameras will be mapped to the center of the nearest grid cell. The direction and speed of the target are determined based on its current and previous locations. The static cameras detect the targets in their fov's and report their locations, directions, and speeds to the MDP controller.

By calibrating the active cameras, the possible target locations in the environment that lie within the fov of each active camera in its various states can be pre-computed. For each state $c_i \in \mathcal{C}$ of active camera i, the subset of locations lying within its corresponding fov is denoted by $fov(c_i) \subset \mathcal{T}_l$. For example, Figure 3.4 illustrates the fov (i.e., shaded polygon) of active camera 1 in its current state c_1; the subset of locations that are observed by camera 1 is given by $fov(c_1) = \{(0,1),(0,2),\ldots,(2,3),(2,4)\}$.

To observe targets with a guaranteed resolution, the zoom parameter of an active camera can be adjusted to focus its fov so that imageries of the targets detected within its fov

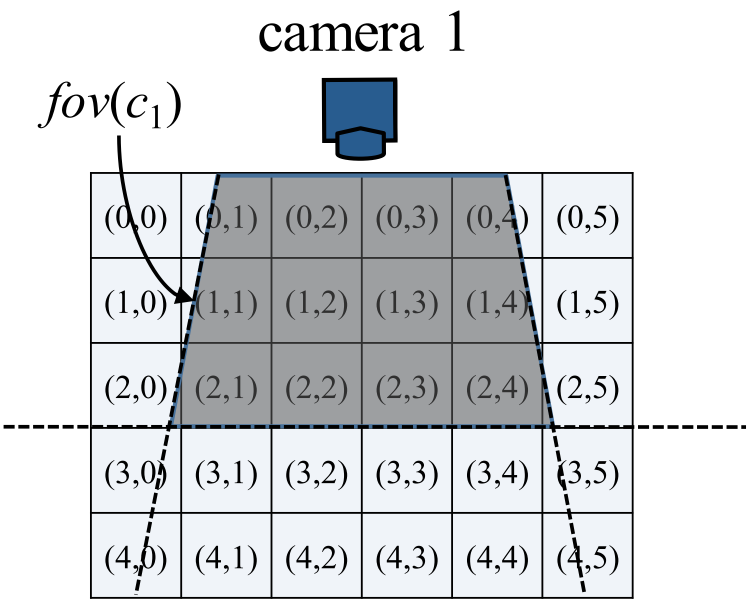


Figure 3.4: $fov(c_1)$ of camera 1.

satisfy a pre-defined resolution. This requires limiting the depth of its fov, as depicted by the horizontal line in Figure 3.4. As a result, if a target is located within $fov(c_i)$ of any camera i, then it is observed with a guaranteed resolution. For example, the minimum resolution of the human face should be 24 × 24 pixels, which is the base resolution for face detection [Viola and Jones, 2004]. The resolution of the targets should be higher than 24 × 24 pixels for other tasks like face recognition and expression analysis, vehicle number plate detection and identification, etc. Let the vector $C = (c_1, c_2, \ldots, c_n)$ be the joint state of n active cameras in the environment and the vector $T = (t_1, t_2, \ldots, t_m)$ be the joint state of m targets in the environment where $t_k \in \mathcal{T}$ is the state of target k. A state $S \in \mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$ of the MDP is therefore of the form $S = (T, C)$.

The actions of an active camera are pan/tilt/zoom commands to move the camera to a specified state. Let a_i be an action of camera i corresponding to a pan/tilt/zoom command. We assume that the delay in moving the camera to a specified state is negligible as the state-of-the-art cameras are capable of panning at a speed of $360°$/sec [Axi, 2011]. The joint action of all cameras at any given time is a vector $A = (a_1, a_2, \ldots, a_n) \in \mathcal{A}$. Since we assume that the targets' motion cannot be controlled, no action can be specified by the MDP controller to influence their motion in the surveillance environment.

3.3.2 Transition Function T_f

Recall that the transition function T_f of the MDP denotes the probability $P(S'|S, A)$ of moving from the current state S to the next state S' using the joint action A. In this subsection, we will show how this transition probability can be factored into transition probabilities of individual active cameras and targets using the conditional independence property, which is inherent in the state transition dynamics of the surveillance environment. As a result, the computation time of our optimal policy is significantly reduced (i.e., from exponential to linear in the number m of targets), hence alleviating the scalability issue (see Theorem 4).

Firstly, the transition probability $P(S'|S, A)$ can be factored into the transition probabilities of the active cameras and targets (i.e., respectively, $P(C'|C, A)$ and $P(T'|T)$) due to conditional independence (see first equality of (3.2)). Specifically, the transition probability $P(C'|C, A)$ of the active cameras is conditionally independent of the targets' states. Since the targets are assumed to be oblivious to the cameras, the transition probability $P(T'|T)$ (i.e., motion model) of the targets is conditionally independent of the active cameras' states and actions.

Next, the transition probability $P(C'|C, A)$ of the active cameras can also be factored into transition probabilities of individual active cameras due to conditional independence. The transition probability of an individual camera i is $P(c'_i|c_i, a_i)$ where $c_i, c'_i \in \mathcal{C}$ are, respectively, its current and next states, and a_i is its action. Since the transition probability of each active camera is conditionally independent of the other cameras given its current state and action, $P(C'|C, A)$ can be factored into $P(c'_i|c_i, a_i)$'s for $i = 1, \ldots, n$ (see second equality of (3.2)). Modern active cameras are equipped with advanced functionalities that enable them to move to the desired pan/tilt/zoom positions accurately [Axi, 2011]. Hence, it is practical to assume the transition of camera i to be deterministic and consequently represented by a deterministic function τ, that moves the camera from its current state c_i to the next state c'_i by the action a_i, i.e., $c'_i = \tau(c_i, a_i)$. Therefore the transition probability $P(c'_i|c_i, a_i)$ evaluates to either 0 or 1.

Similarly, the transition probability $P(T'|T)$ of the targets can be factored into transition probabilities (i.e., motion models) of individual targets by assuming conditional independence. The transition probability of target k is $P(t'_k|t_k)$ where $t_k, t'_k \in \mathcal{T}$ are, respectively, its current and next states. Since the transition probability of each target is conditionally independent of the other targets given its current state, $P(T'|T)$ can be factored into $P(t'_k|t_k)$'s for $k = 1, \ldots, m$ (see second equality of (3.2)). Although this assumption does not hold in practice, we gain significant computational efficiency and performance by having this assumption in our coordination framework.

As discussed above, the transition probability $P(S'|S, A)$ of the MDP can be factored into transition probabilities of individual active cameras and targets after repeatedly applying the conditional independence property:

$$\begin{aligned} P(S'|S,A) &= P(C'|C,A)\; P(T'|T) \\ &= \prod_{i=1}^{n} P(c'_i|c_i,a_i) \prod_{k=1}^{m} P(t'_k|t_k) \\ &= \begin{cases} \displaystyle\prod_{k=1}^{m} P(t'_k|t_k) & \text{if } P(c'_i|c_i,a_i) = 1 \text{ for } i = 1,\ldots,n, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3.2}$$

3.3.2.1 Transition Probability $P(t'_k|t_k)$ of a Target

To calculate the transition probability of a target, we first predict a target's movement in a surveillance environment using a general direction-speed motion model [Thrun *et al.*, 2005; Bruce and Gordon, 2004; Hightower and Borriello, 2004]. Specifically, this model comprises two Gaussian distributions for the speed v and direction d of the target: $v \sim \mathcal{N}(\mu_v, \sigma_v)$ and $d \sim \mathcal{N}(\mu_d, \sigma_d)$ where the mean parameters μ_v and μ_d are obtained from the static cameras at every time step based on the previous location of the target, and the variance parameters σ_v and σ_d are learned from a dataset of targets' trajectories in the given supervised surveillance environment.

Then, in every time step t, we draw paired samples of speed v and direction d of the target from the Gaussian distributions, compute its corresponding predicted location (x_t, y_t) in the environment using

$$\begin{aligned} x_t = &\quad x_{t-1} + v \times \cos(d) \times \mathrm{d}t \\ y_t = &\quad y_{t-1} + v \times \sin(d) \times \mathrm{d}t \end{aligned} \tag{3.3}$$

and determine the proportion of samples in each grid cell to produce the transition probability $P(t'_k|t_k)$ of the target. Figure 3.5 shows the transition probability distribution of a target that is located at $(x_{t-1}, y_{t-1}) = (5, 5)$ with $\mu_v = 2$ cells per time step and $\mu_d =$ $45°$. The probability distribution of the neighboring locations that the target will move to in time step t is shown as black dots. Since the possible locations, directions, and speeds of the target are finite, we can pre-compute the transition probabilities of the target and store them off-line. This helps to reduce the on-line policy computation time, as discussed in Theorem 2.

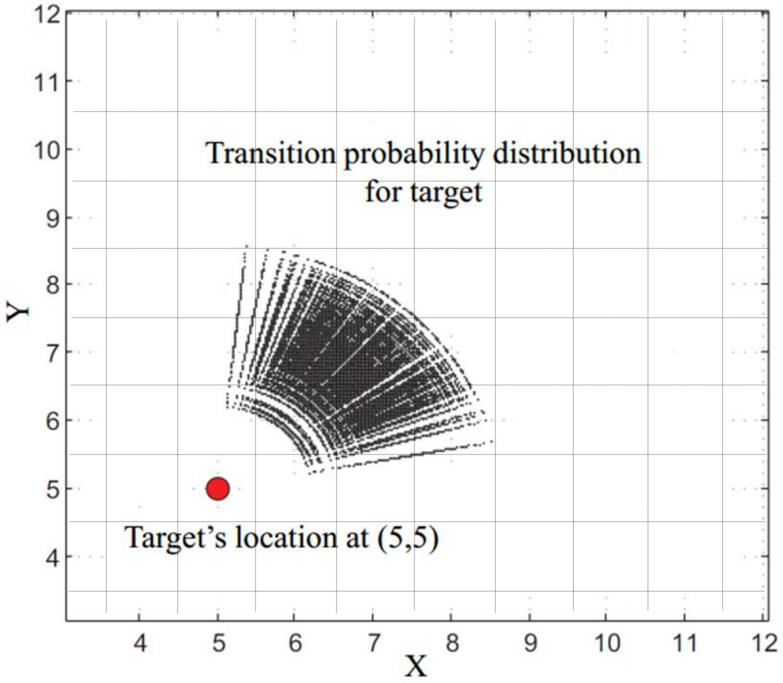


Figure 3.5: Transition probability distribution of a target.

3.3.3 Objective/Reward Function R

The advantage of using MDPs in surveillance systems is that any high-level surveillance goal can be defined formally using a real-valued objective/reward function. In this work, the goal of the surveillance system is to maximize the number of observed targets with a guaranteed resolution. Supposing the states of all targets are known, such a goal can be achieved by defining a reward function that measures the total number of targets lying within the fov of any of the active cameras:

$$R(S) = R((T, C)) \triangleq \sum_{k=1}^{m} \widetilde{R}(t_k, C) \tag{3.4}$$

$$\widetilde{R}(t_k, C) \triangleq \begin{cases} 1 & \text{if target } k\text{'s location lies in } fov(C), \\ \\ 0 & \text{otherwise;} \end{cases} \tag{3.5}$$

where $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$ denotes a set of target locations in the environment, each of which lies within the fov of at least one active camera when the cameras are in state C. So, if the location of target k lies within $fov(C)$, then it is guaranteed to be observed at a predefined image resolution, as discussed in Section 3.3.1, and $\widetilde{R}(t_k, C) = 1$ results.

3.4 Policy Computation

3.4.1 Greedy policy

The states of the targets in the next time step are uncertain due to stochasticity of their motion. Therefore, the optimal greedy policy π_g^* (i.e., one step look-ahead) has to instead maximize the *expected* total number of targets that lie within the fov of any of the active cameras in the next time step:

$$\pi_g^*(S) = \pi_g^*((T, C)) = \arg\max_{A \in \mathcal{A}} V(T, C, A). \tag{3.6}$$

The value function for the greedy case is defined as follows:

$$V(T, C, A) = \sum_{T' \in \mathcal{T}^m} R((T', C'))\ P(T'|T) \tag{3.7}$$

where T' and C' are, respectively, the joint states of the targets and active cameras in the next time step. The next joint state C' of the cameras can be determined deterministically from their current joint state C and action A using the function $c_i' = \tau(c_i, a_i)$ for $i = 1, \ldots, n$ (Section 3.3.2).

Computing the policy π^* (3.6) for a given state S incurs $\mathcal{O}(|\mathcal{A}||\mathcal{T}|^m)$ time, which is exponential in the number m of targets[1]. Its time complexity can be significantly reduced by exploiting the inherent structure of our surveillance problem, in particular, the conditional independence property in the transition model of the MDP (Section 3.3.2.1). As a result, the value function V (3.7) of joint state of m targets can be reduced to sum of value functions $\widetilde{V}$ of individual targets as follows:

$$V(T, C, A) = \sum_{k=1}^{m} \widetilde{V}(t_k, C') \tag{3.8}$$

$$\widetilde{V}(t_k, C') \triangleq \sum_{t'_k \in \mathcal{T}} \widetilde{R}(t'_k, C')\ P(t'_k|t_k)\ . \tag{3.9}$$

For a detailed derivation of (3.8), see Appendix A.1. Computing the policy π^*_g for a given state S consequently incurs linear time in the number m of targets, as shown in the result below:

Theorem 1. *If (3.2) holds, then computing policy π^*_g (3.6) for a given state S incurs $\mathcal{O}(|\mathcal{A}||\mathcal{T}|m)$ time.*

To improve the real-time computation of policy π^*_g, the values of $\widetilde{V}(t_k, C')$ (3.9) for all $t_k \in \mathcal{T}$ and $C' \in \mathcal{C}^n$ can be pre-computed and stored off-line. To do this, the values of $P(t'_k|t_k)$ for all $t_k, t'_k \in \mathcal{T}$ have to be pre-computed first, which incurs $\mathcal{O}(|\mathcal{T}|^2)$ time. The values of $\widetilde{R}(t'_k, C')$ for all $t'_k \in \mathcal{T}$ and $C' \in \mathcal{C}^n$ also have to be pre-computed, which incurs $\mathcal{O}(|\mathcal{T}||\mathcal{C}|^n)$ time. Consequently, the values of $\widetilde{V}(t_k, C')$ (3.9) for all $t_k \in \mathcal{T}$ and $C' \in \mathcal{C}^n$ can be pre-computed in $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|^n)$ time. Hence, the total off-line computation time is $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|^n)$. The on-line computation time to derive policy π^*_g can then be reduced to $\mathcal{O}(|\mathcal{A}|m)$, which includes the time taken to look up the values of $\widetilde{V}(t_k, C')$ for m targets (3.8) and over $|\mathcal{A}|$ possible joint actions (3.6). The result below summarizes the

[1] The number of actions of MDP $|\mathcal{A}|$ is still exponential in number of cameras and hence our work has its limitation on scalability in number of cameras.

computation time incurred by the on-line and off-line processing steps:

Theorem 2. *If (3.2) holds, then computing policy π_g^* (3.6) for a given state S incurs off-line computation time of $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|^n)$ and on-line computation time of $\mathcal{O}(|\mathcal{A}|m)$.*

3.4.2 Planning Policy

The policy for a given state $S \in \mathcal{S}$ associated with finite h-horizon planning (i.e., more than one step look-ahead) is given by,

$$\pi_h^*(S) \quad = \quad \pi_h^*((T,C)) = \arg\max_{A \in \mathcal{A}} V^h(T,C,A) \tag{3.10}$$

where the value function $V^h(T,C,A)$ is defined as follows:

$$V^h(T,C,A) = \sum_{T' \in \mathcal{T}^m} \left[\underbrace{R((T',C'))}_{expected\ reward} + \underbrace{\gamma \max_{A' \in \mathcal{A}} V^{h-1}(T',C',A')}_{future\ reward} \right] P(T'|T). \tag{3.11}$$

where γ is the discount factor and $0 < \gamma < 1$.

It can be seen that for our MOMT problem, the finite horizon planning policy and the greedy policy are equivalent, i.e., the finite planning policy $\pi_h^*(S)$ and the greedy policy $\pi^*(S)$ for the given state $S \in \mathcal{S}$, returns the same optimal action A. This is due to the deterministic state transition of active cameras and their action space. The detailed proof is shown below:

Proof: *Proof by induction* on h that $\pi_h^*(S) = \pi_g^*(S)$ for finite positive value of h.

Base case ($h = 1$): Greedy policy is a special case of finite horizon planning policy when $h = 1$, i.e., $\pi_1^*(S) = \pi_g^*(S)$.

Consider,

$$\pi_1^*(S) = \arg\max_{A \in \mathcal{A}} V^1(T,C,A)$$

where the value function $V^1(T, C, A)$ for horizon $h = 1$ is given by,

$$\begin{aligned} V^1(T, C, A) &= \sum_{T' \in \mathcal{T}^m} \left[R((T', C')) + \gamma \max_{A' \in \mathcal{A}} V^0(T', C', A') \right] P(T'|T) \\ &= \sum_{T' \in \mathcal{T}^m} R((T', C'))P(T'|T) \\ &= V(T, C, A). \end{aligned}$$

The second equality follows because $V^0(T', C', A') = 0$ for all $A' \in \mathcal{A}$. Therefore, the base case $\pi_1^*(S) = \pi_g^*(S)$ is true.

Base case ($h = 2$**):** We have to prove that $\pi_2^*(S) = \pi_g^*(S)$, given $\pi_1^*(S) = \pi_g^*(S)$ is true. Consider,

$$\pi_2^*(S) = \arg\max_{A \in \mathcal{A}} V^2(T, C, A)$$

where the value function $V^2(T, C, A)$ for horizon $h = 2$ is given by,

$$V^2(T, C, A) = \sum_{T' \in \mathcal{T}^m} \left[R((T', C')) + \gamma \max_{A' \in \mathcal{A}} V^1(T', C', A') \right] P(T'|T)$$

where T and C are the current joint states of targets and cameras. As stated previously, the next joint state C' of the cameras can be determined deterministically from their current joint state C and action A using the function $c_i' = \tau(c_i, a_i)$ for the cameras $i = 1, \ldots n$. The value function $V^1(T', C', A')$ is given by,

$$V^1(T', C', A') = \sum_{T'' \in \mathcal{T}^m} R((T'', C''))P(T''|T').$$

Similarly, the new state of the cameras C'' can be determined from C' and A' by the deterministic function $c_i'' = \tau(c_i', a_i')$ for the cameras $i = 1, \ldots, n$.

It can be seen from the value functions V^2 and V^1, that the action A in V^2 is independent of the future rewards, i.e., $\forall A \in \mathcal{A}$ in V^2, the future reward $\max\limits_{A' \in \mathcal{A}} V^1(T', C', A')$ returns the same value. This is due to:

1. The state transition of active cameras $P(C'|C, A)$ in V^2 is deterministic and is given by the deterministic function $c_i' = \tau(c_i, a_i)$ for the cameras $i = 1, \ldots, n$.
2. There exists a joint action $A \in \mathcal{A}$ for the cameras to move from any of the current states $C \in \mathcal{C}^n$ to any of the next states $C' \in \mathcal{C}^n$, i.e., there is always an action a_i for a camera i, such that it can pan/tilt/zoom to any of the next states $c'_i \in \mathcal{C}$ from any of its current states $c \in \mathcal{C}$ by the function $c'_i = \tau(c_i, a_i)$.
3. The above two reasons are due to the practical assumption that the active cameras can move accurately and at faster rate, as todays active cameras can pant/tilt/zoom at the rate of 360°/sec [Axi, 2011].

The future reward term $\max\limits_{A' \in \mathcal{A}} V^1(T', C', A')$ is constant with respect to the action A in V^2. Therefore, for all the action $A \in \mathcal{A}$ in calculating the policy $\pi_2^*(S)$, the future reward is constant and only the expected reward $R((T', C'))$ varies, which implies $\pi_2^*(S) = \pi_g^*(S)$. This shows that the base case for $h = 2$ is also true.

Inductive case: Suppose that, $\pi_{i-1}^*(S) = \pi_g^*(S)$ for $h = (i-1)$ is true for some finite value of i such that $h > 2$. We have to show that $\pi_i^*(S) = \pi_g^*(S)$ is also true. The policy for $h = i^{th}$ horizon is given by,

$$\pi_i^*(S) = \arg\max_{A \in \mathcal{A}} V^i(T, C, A)$$

where the value function $V^i(T, C, A)$ for horizon $h = i$ is given by,

$$V^i(T, C, A) = \sum_{T' \in \mathcal{T}^m} \left[R((T', C')) + \gamma \max_{A' \in \mathcal{A}} V^{i-1}(T', C', A') \right] P(T'|T).$$

In the above equation, $\forall A \in \mathcal{A}$ in V^i, the future reward term $\max\limits_{A' \in \mathcal{A}} V^{i-1}(T', C', A')$ is a constant with respect to A. This is because, the current action A does not influence the future reward term as stated in the base case $(h = 2)$. Therefore, for all $A \in \mathcal{A}$, the policy $\pi_i^*(S)$ also returns the same future rewards and varies only the expected reward $R((T', C'))$ which implies $\pi_i^*(S) = \pi_g^*(S)$ is also true. Therefore, for our MOMT problem, planning actions for active cameras over finite horizon step is equivalent to the greedy solution, which is one step look-ahead solution.

3.5 Experiments and Discussion

In this section, we present empirical evaluation of our MDP-based approach (given by (3.6)) for maximizing the number of targets observed by active cameras. Our proposed approach is simulated in Player/Stage simulator [Gerkey *et al.*, 2003] to perform extensive experimentations and implemented using real Axis 214 PTZ cameras to demonstrate its feasibility in real surveillance system. Before describing them, it is important to point out that there is no standard benchmark surveillance environments and datasets for active camera networks to compare our proposed approach with the other systems in the literature (e.g., [Krahnstoever *et al.*, 2008; Qureshi and Terzopoulos, 2009; Sommerlade and Reid, 2010]). While the primary criterion of these systems is to focus and track one or more targets at high-resolution, our objective function is to maximize the number of targets observed in high-resolution images (see Table 2.2). These existing systems use heuristic approaches that can optimize only their respective objective function and cannot be used for other objective functions. These systems also suffer from scalability issue when the number of targets is increased. Therefore our MDP-based approach (denoted as MDP in the Figures 3.8, 3.9 and 3.10) is empirically compared with the following existing heuristic methods:

- *Without prediction of targets' trajectories (WoP)*: The active cameras are controlled to

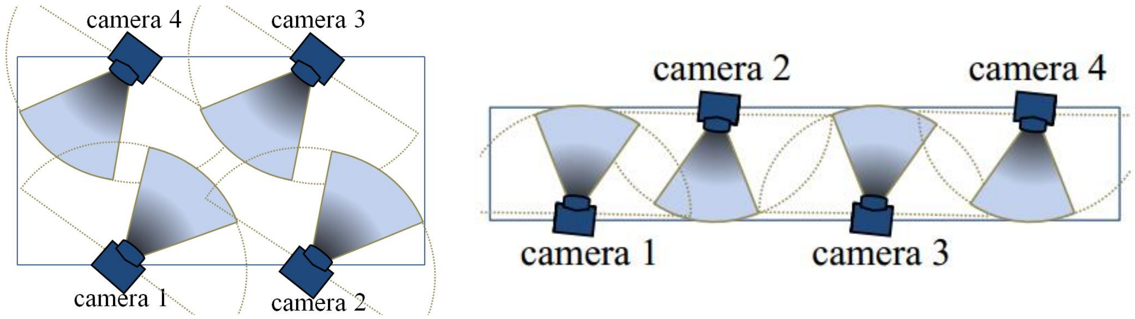


Figure 3.6: Setups of corridor and hall environments in $Stat$ approach.

observe the targets based on the location of the targets in the current time step rather than their predicted locations in the next time step;

- *Systematic Approach (Sys)* : The active cameras pan automatically in a round robin fashion such that every camera pans to each of its states for a finite duration;
- *Static Approach (Stat)*: The active cameras are fixed at specific states such that they can cover maximum area to get high-resolution imageries of the targets (see Figure 3.6).

Our approach and the above heuristic methods are evaluated using the following performance metric:

$$PercentObs = \frac{100}{\tau M_{tot}} \sum_{i=1}^{\tau} M_{obs}^{i}$$

where τ (i.e., set to 100 in simulations) is the total number of time steps taken in our experiments, M_{obs}^{i} is the total number of targets observed by the active cameras at a given time step i, and M_{tot} is the total number of targets present in the environment. That is, the $PercentObs$ metric averages the percentage of targets being observed by the active cameras over the entire duration of τ time steps. We will first discuss the environmental setup for the simulated experiments and analyze the experimental results. Then, we will show the results of the real camera experiments. Interested readers can view our demo video[2].

[2] http://www.comp.nus.edu.sg/~lowkh/camera.html

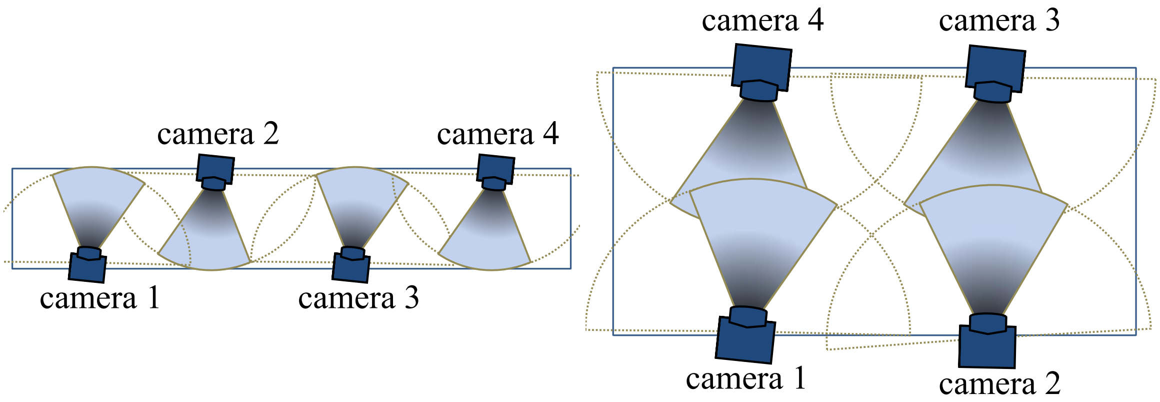
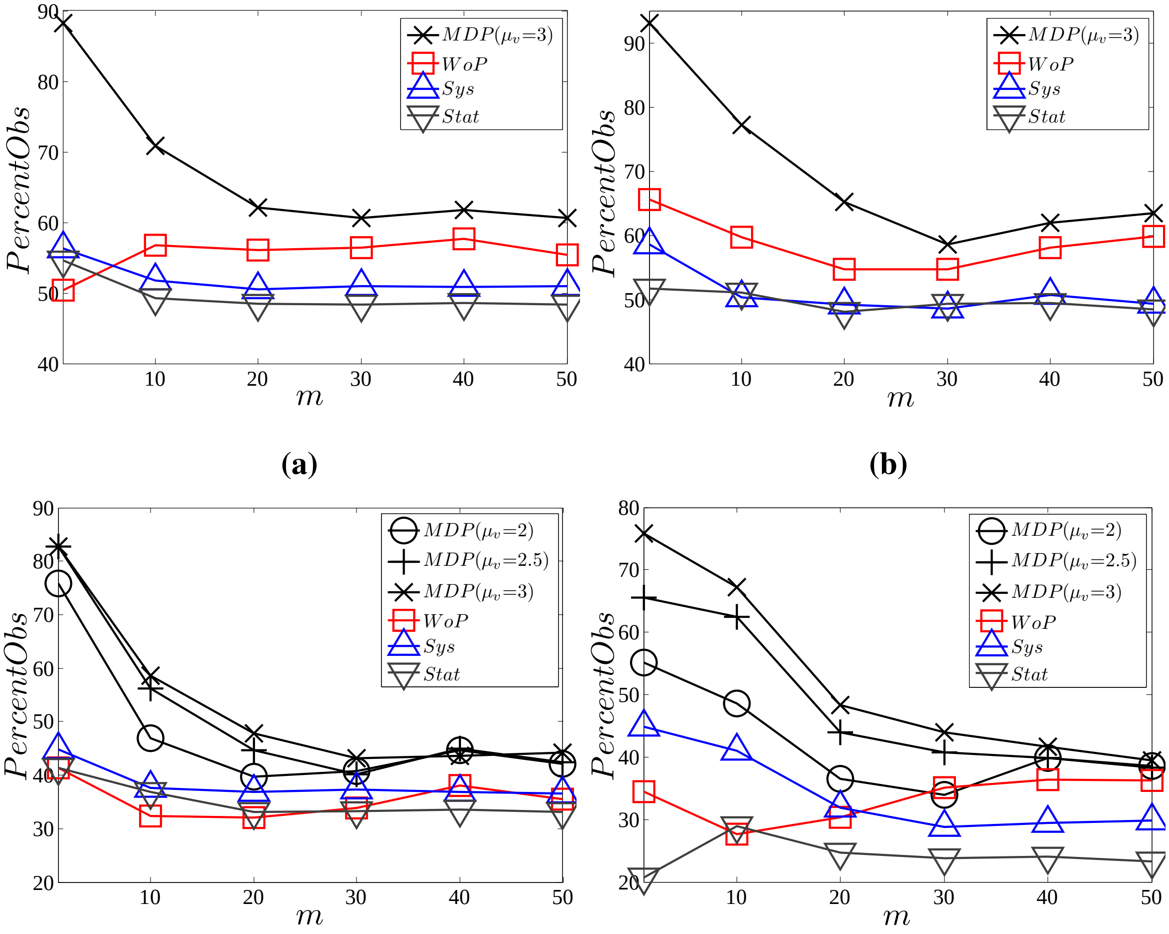


Figure 3.7: Setups of corridor and hall environments.

3.5.1 Simulated Experiments: Setup

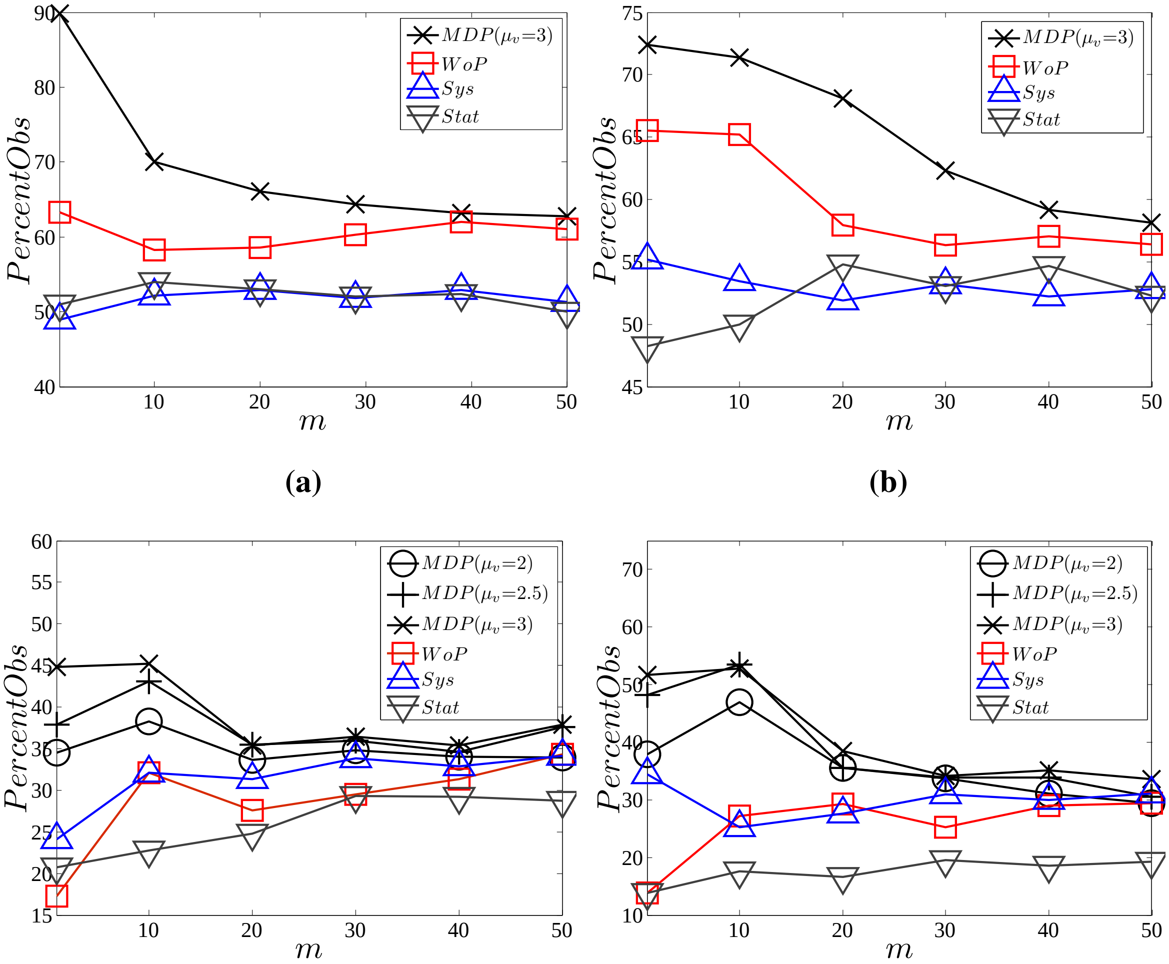
In Player/Stage simulator, we have designed an active camera model with functionalities to simulate real active cameras by configuring the number of states across pan angles, as discussed in Section 3.3.1. Targets' trajectories are generated automatically using the Player/Stage simulator based on direction-speed motion model (see (3.3)). In order to make the trajectories more stochastic, we manually draw the trajectories in the simulator. The locations of the targets are determined by a static camera, which is the simulator itself. We have conducted our experiments for two environmental setups (Figure 3.7): corridor and hall. The sizes of the corridor and hall environments are, respectively, 40×5 grid cells and 20×10 grid cells such that $|\mathcal{T}_l| = 200$. The size of a grid cell in the simulator is approximately mapped to 1 m^2 in real world. We have used up to $n = 4$ active cameras with $|\mathcal{C}| = 3, 5$, and tested up to $m = 50$ targets. We have also conducted experiments for the camera resolutions $|fov(c_i)| \approx 25, 16$ by reducing the size of the camera's fov polygon in the simulator. The set $fov(c_i)$ of target locations that are observed by each active camera is determined by calibrating the active cameras in each of its state.



(a) **(b)**

(c) **(d)**

Figure 3.8: Graphs of $PercentObs$ vs. number m of targets for *corridor setup*: (a) non-clustered targets with $|fov(c_i)| \approx 25$ cells, $|\mathcal{C}| = 3$ and clustered targets with (b) $|fov(c_i)| \approx 25$ cells, $|\mathcal{C}| = 3$, (c) $|fov(c_i)| \approx 16$, $|\mathcal{C}| = 3$, (d) $|fov(c_i)| \approx 16$, $|\mathcal{C}| = 5$.



(a) **(b)** **(c)** **(d)**

Figure 3.9: Graphs of $PercentObs$ vs. number m of targets for *hall setup*: (a) non-clustered targets with $|fov(c_i)| \approx 25$ cells, $|\mathcal{C}| = 3$ and clustered targets with (b) $|fov(c_i)| \approx 25$ cells, $|\mathcal{C}| = 3$, (c) $|fov(c_i)| \approx 16$, $|\mathcal{C}| = 3$, (d) $|fov(c_i)| \approx 16$, $|\mathcal{C}| = 5$.

3.5.2 Simulated Experiments: Results

Figures 3.8 and 3.9 show the performance of our MDP-based approach for corridor and hall setups with $n = 4$, $|\mathcal{T}_l| = 200$, mean of the target's speed $\mu_v = 3$ cells per time step, and with varying m, $|fov(c_i)|$, $|\mathcal{C}|$, and sizes of clusters of targets that follow the Poisson distribution ($\lambda = 3$). The rest of this subsection describes the observations from our experiments.

Our MDP-based approach performs better for any $\mu_v = 1, 2$, and 3 cells per time step. This is because the cameras are controlled based on the predicted locations of a target by matching its corresponding transition probabilities with respect to its observed state. It can be observed from the experiments that the performance of MDP is much better that the other approaches when (a) the speed of the targets is higher, (b) the targets move in clusters, and (c) when the resolution of the cameras is increased (i.e., $|fov(c_i)|$ is decreased). This is because when the speed of the targets is high (i.e., $\mu_v = 2.5, 3$ cells per time step), all the targets will almost certainly move out of the fov's of the cameras in WoP approach as the cameras are controlled based on the current location of the targets, hence producing worse performance (see Figures 3.8(a) and 3.9(a)). When the targets move in clusters, then the WoP approach suffers even more performance degradation because it has high tendency to lose clusters of targets. On the other hand, since the MDP has the correct transition model, it gives superior performance even when the targets move in high speed. By increasing the resolution of the active cameras (i.e., by reducing $|fov(c_i)| \approx 25$ to 16), it can be observed that the MDP performs much better when compared to the WoP approach (Figures 3.8(c), 3.8(d), 3.9(c), and 3.9(d)). This is because when the targets are moving at a speed whose $\mu_v = 3$ cells per time step and are observed at higher resolution (i.e., $|fov(c_i)|$ is smaller), the chance of losing the targets is high when the cameras are controlled based on current observed locations of the targets. Since MDP has transition model that predicts the next locations of the targets, it outperforms the other approaches when the targets are clustered and the resolution of the cameras is high.

The Sys and $Stat$ approaches perform worse in almost all cases except when $|fov(c_i)| \approx 16$ (Figures 3.8(c), 3.8(d), 3.9(c), and 3.9(d)) where the Sys approach performs better than WoP approach. This is due to the fact that the cameras are controlled independently of the targets' information in both Sys and $Stat$ approaches. This shows that the targets' information (e.g., location, direction, etc) play a vital role in achieving high-quality surveillance. But, when $|fov(c_i)| \approx 16$, the Sys approach performs slightly better than WoP because the chance of targets moving out of the fov is higher in WoP approach if $\mu_v = 3$ cells per time step and the fov is reduced to $|fov(c_i)| \approx 16$. In all cases, the MDP outperforms the Sys and $Stat$ approaches.

When the number of states of each camera is increased from $|\mathcal{C}| = 3$ (Figures 3.8(c) and 3.9(c)) to $|\mathcal{C}| = 5$ (Figures 3.8(d) and 3.9(d)), the performance improves because more targets can be observed due to the additional camera states. The MDP-based approach performs better than the other approaches even when the transition model is inaccurate. This is tested by moving the targets at a speed whose $\mu_v = 3$ cells per time step and matching the transition probabilities computed with speeds whose $\mu_v = 2, 2.5, 3$ cells per time step (Figures 3.8(c), 3.8(d), 3.9(c), and 3.9(d)). The performance of MDP computed with inaccurate transition probabilities is still much better than the other approaches. This is because the reward function is optimized with respect to the expected locations of the targets.

When the number of cameras is increased from $n = 2, 3$ to 4, the increase in performance of MDP is much better than the other approaches for $m < 10$ targets and comparable to (if not better than) other approaches for $m > 10$. This is because the prediction capability of our approach outperforms the other approaches with every addition of a new camera. The graph with increasing number of cameras is shown in Figure 3.10.

From these observations, we find that our MDP-based approach performs better than the other tested approaches in all the cases due to its prediction capability. Specifically, it

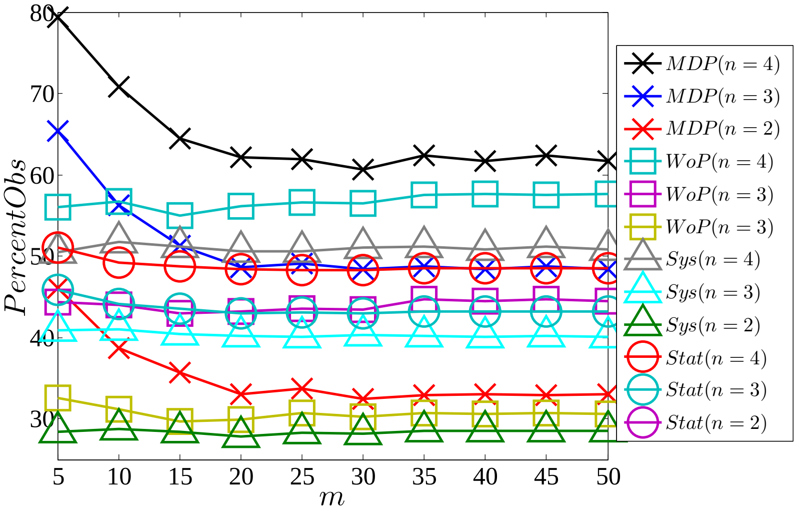
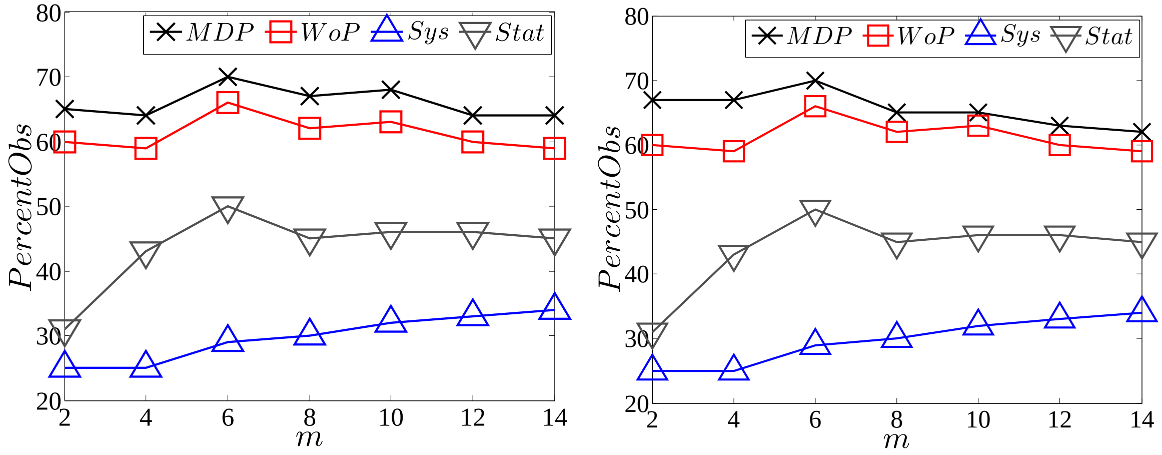


Figure 3.10: Graphs of $PercentObs$ vs. number m of targets in *corridor setup* for number of active cameras $n = 2, 3$ and 4.

outperforms WoP approach when the speed of the targets and the resolution of the cameras are high.

In order to test our MDP framework with real human trajectories, we have used standard PETS dataset [pet, 2013] and manually extracted the human trajectories for $m = 14$ targets. We have created the simulation environment in the Player/Stage simulator based on the ground truth data from the dataset and virtually setup up to $n = 4$ active cameras each with $|\mathcal{C}| = 8$ states in that environment. Figure 3.11(a) shows the results of MDP framework with the real human trajectory data and the target's transition probabilities are computed based on direction-speed motion model (see Section 3.3.2.1). Figure 3.11(b) shows the results of our MDP framework for the same human trajectory data where the target's transition probabilities are computed based on Brownian motion model. Our MDP based approach outperforms other approaches in this setup with the real human trajectory data.



(a)

(b)

Figure 3.11: Graphs of $PercentObs$ vs. number m of targets for real human trajectories from PETS dataset with transition probabilities of targets calculated based on (a) direction-speed motion model and (b) Brownian motion model.

3.5.3 Real Camera Experiments

We have conducted real experiments with $n = 3$ Axis 214 PTZ cameras to monitor up to $m = 6$ Lego robots (targets) in an environment with the size of $|\mathcal{T}_l| = 11 \times 9$ grid cells. The size of each grid cell is 0.5 m^2. Each camera has $|\mathcal{C}| = 3$ states. The states of the cameras are determined such that all the cells of the environment can be observed at high-resolution by at least one camera. Given any joint state C of the cameras, only a subset of cells in the environment can be observed by these cameras, i.e., $fov(C) \subset \mathcal{T}_l$. This makes the problem challenging for the active cameras to maximize the number of observed robots. We have a static camera that can track these robots based on OpenCV Camshift tracker. The static camera is calibrated using [Tsai, 1986] to obtain the approximate locations of the robots at every time step. The direction and speed of the robots are determined based on their previous and current locations. The $fov(C)$ is determined by calibrating the active cameras in each of its state and determining the grid cells of the environment in which the robots can be observed at a high-resolution. We guarantee the resolution of the robots that are observed by the active cameras to be approximately more than 40×40 pixels. We pre-computed the transition probabilities of an individual target for all possible locations, directions, and speeds whose $\mu_v = 1$ and 2 cells per time step. The robots are moved based on the direction-speed motion model and are programmed to turn back or stop when they hit the wall or cross other robots. Each robot is initialized with a Camshift tracker in the static camera and is tracked to get its approximate 3D location, direction, and speed.

We have tested our implementation up to $m = 6$ robots but we keep one of the robots static. It can be observed that cameras 2 and 3 coordinate to observe the brown static robot (Figure 3.12). Camera 2 pans to another state (see bottom two rows of Figure 3.12) only when camera 3 takes over the observation of the static target (see top two rows of Figure 3.12). This static target can be replaced by a portion of the surveillance environment like the entrance/exit or reception where we need to pay more attention. Table 5.2 shows the $PercentObs$ performance for the real experiments over $\tau = 50$ time steps.

Table 3.2: Performance of MDP framework in real camera experiments.

Number m of targets	1	2	3	4	5	6
$PercentObs$	99.2	97	95.3	93.5	88	85.1

3.6 Summary

In this chapter, a novel decision-theoretic approach to control and coordinate multiple active cameras for **maximizing the observations of multiple targets (MOMT)** under *fully observable* surveillance environment has been presented. Specifically, it utilizes the Markov Decision Process framework, which accounts for the stochasticity of targets' motion via a probabilistic motion model and addresses the trade-off by maximizing the expected number of observed targets with a guaranteed resolution via stochastic optimization. The conditional independence property, which is inherent in our surveillance problem, is exploited in the transition model of the MDP to reduce the exponential policy computation time to linear time. As shown in simulations, our approach can scale up to 50 targets in real-time. We have also implemented our proposed decision-theoretic approach using real Axis 214 PTZ cameras to demonstrate its feasibility in real surveillance system. The MDP framework presented in this chapter is scalable only in number of targets, but incurs exponential computation time for increasing number of cameras. Therefore, we address this scalability issue in number of cameras in Chapter 4.

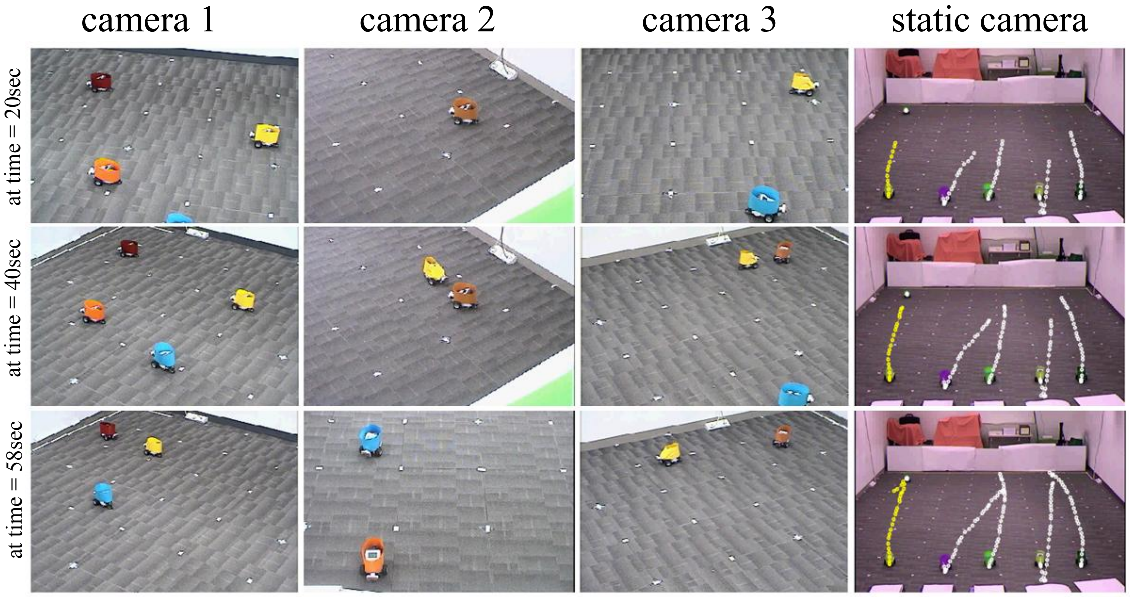


Figure 3.12: Results of real experiments: columns 1 to 3 show the high-resolution images of Lego robots captured by cameras 1, 2, and 3 while column 4 shows the targets' trajectories tracked by the static camera.

Chapter 4

MOMT Problem using Coordination Graph and Max-plus Algorithm

4.1 Introduction

This chapter presents a novel decision-theoretic approach to control and coordinate active cameras for **maximizing the observations of multiple targets (MOMT)** in large-scale *fully observable* surveillance environment. In Chapter 3, we have addressed the scalability in number of targets, but the policy computation incurs exponential computational time when the number of cameras increases. In this chapter, we extend our MDP framework in Chapter 3 using the concepts of coordination graphs (CG) [Guestrin *et al.*, 2002] and max-plus algorithm [Vlassis *et al.*, 2004; Kok and Vlassis, 2006] in order to address this scalability issue.

In addition to the issues addressed in Chapter 3, there are other practical issues in controlling and coordinating PTZ cameras that are specific to large-scale active camera network: (a) each camera should have a mechanism to gather information from relevant cameras to aggregate the knowledge about the decisions of other cameras and make their decisions locally, (b) It is important to reduce the communication overhead between the cameras during course of making decisions. (c) Importantly, the coordination framework should be fault-tolerant to camera/communication failures. When there is a failure of a camera or a communication link between cameras, then the performance of entire camera network should not be affected. This is important, especially when there are many cameras that are distributed across the environment.

In this chapter, we apply the concepts of CG (Section 4.3.4) and max-plus algorithm (Section 4.3.5) to extend our MDP framework (Section 4.2) in Chapter 3 to address the scalability issue in number of cameras. In max-plus algorithm, each camera repeatedly exchange messages with their neighboring cameras and computes its individual optimal action locally over the received messages. Since the computation is distributed to individual cameras rather than to a centralized controlled in Chapter 3, our approach can improve the scalability in number of cameras. This approach addresses the above mentioned issues in the following ways: (a) The max-plus algorithm operates on CG which integrates the coordination structure of underlying active camera network. Each camera, which is a node in the CG, exchanges messages only with its immediate neighboring cameras without broadcasting to other cameras in the network. (b) A message that a camera sends/receives is a locally optimized payoff function and the size of the messages exchanged is very small when compared to the size of image frames. (c) In max-plus algorithm, each camera computes its optimal action on its received messages. When there is a failure of a camera or a communication link between cameras, then it will not affect the entire camera network, instead there is a trade-off in quality of solution produced. We demonstrate empirically through simulations, that our proposed approach can scale up to 60 targets and 50 cameras in real-time (Section 4.4).

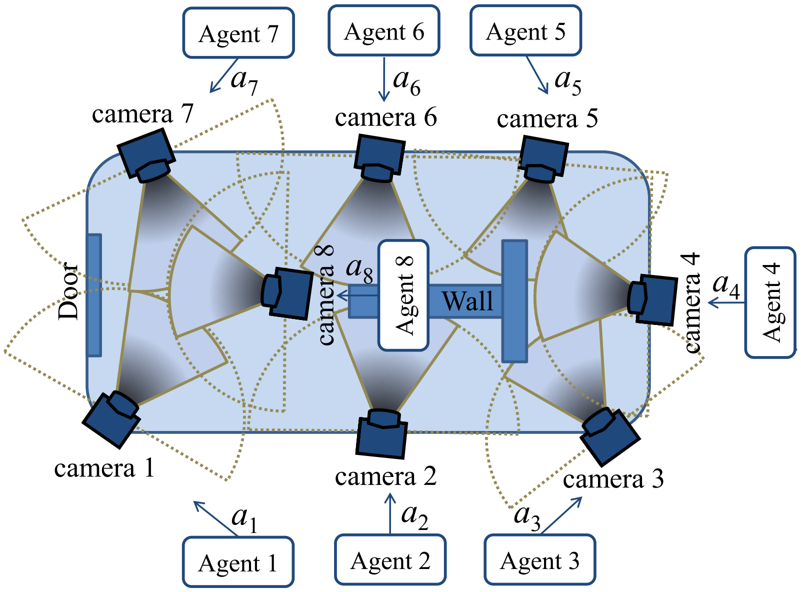


Figure 4.1: System overview for distributed active camera networks. (The static cameras are assumed to be placed at the ceiling and are not shown in the figure for simplicity.)

4.2 System Overview

The proposed framework consists of a supervised surveillance environment and a set of camera controllers known as agents. The surveillance environment consists of targets, active cameras and static cameras. Each active camera is controlled by an agent. The number n of active cameras is much lesser when compared to the number m of targets, i.e., $n \ll m$. The physical surveillance environment is same as described in Chapter 3. In our previous setup, we have a centralized camera controller that computes optimal joint actions of the active cameras. When the number of cameras increases, the joint action space increases exponentially, and hence computing optimal joint action becomes intractable. Whereas in this framework, individual agents will compute the actions only for the respective active camera through message passing. Figure 4.1 shows the overview of the proposed surveillance system with 8 active cameras. We redefine the MDP framework from Chapter 3 as a tuple $(\mathcal{S}, \mathcal{A}, R, T_f)$ that consists of:

- a set $\mathcal{S}$ of discrete states of targets and the active cameras in the surveillance environment (Section 4.3.1),
- a set $\mathcal{A}$ of joint actions of active cameras such that $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n$, where $\mathcal{A}_i$ is the action space of camera i and $a_i \in \mathcal{A}_i$ is the action of camera i (Section 4.3.1),
- a transition function $T_f : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denoting the probability $P(S'|S, A)$ of switching from the current state $S \in \mathcal{S}$ to the next state $S' \in \mathcal{S}$ using the joint action $A \in \mathcal{A}$ (Section 4.3.2),
- a real-valued reward function $R : S \to \mathbb{R}$ representing the high-level surveillance goal (Section 4.3.3).

The global value function $V(S, A)$ and the optimal policy π^* of the MDP framework defined in Chapter 3 (3.7) is given by

$$V(S, A) = V(T, C, A) = \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} \widetilde{R}(t'_k, C')\; P(t'_k | t'_k) \tag{4.1}$$

$$\pi^*(S) = \underset{A \in \mathcal{A}}{argmax}\, V(S, A). \tag{4.2}$$

The joint action space $\mathcal{A}$ is exponential in number of cameras and computing optimal joint action in (4.2) incurs exponential computation time and is computationally intractable for large number of cameras. For example, when there are $n = 50$ active cameras with $|\mathcal{A}_i| = 8$ actions, then the joint action space is $8^{50} = 1.43 \times 10^{45}$ which is computationally expensive and intractable. Therefore, we decompose the centralized coordination problem into individual local sub-problems using the concept of CG and solve the coordination problem approximately using max-plus algorithm.

4.3 Problem Formulation

The problem is to coordinate a set of active cameras to maximize the expected number of targets observed with guaranteed resolution in large-scale camera networks. In this section, we discuss each component of the MDP framework and how we decompose the value function of the centralized MDP controller into sum of value functions of individual agents using CG. Later, we show how the value functions of individual agents are used to compute the approximate optimal actions using the max-plus algorithm. The notations and symbols used in this chapter is summarized in Table 4.1.

Table 4.1: Mathematical notations and its descriptions used in Chapter 4.

Notation	Description
n	Number of active cameras.
m	Number of targets to be monitored by n active cameras, such that $n \ll m$.
$\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$	State space of a target represented by a set of tuples of location, direction and speed.
$\mathcal{T}_l$	Set of all possible discretized locations of the target in the surveillance environment.
$\mathcal{T}_d$	Set of all possible discretized directions of target.
$\mathcal{T}_v$	Set of discretized speeds of the target.
t_k=$(t_{l_k}, t_{d_k}, t_{v_k})$	State of a target k is a tuple consisting of its location $t_{l_k} \in \mathcal{T}_l$, direction $t_{d_k} \in \mathcal{T}_d$ and speed $t_{v_k} \in \mathcal{T}_v$.
$T \in \mathcal{T}^m$	Joint state of m targets represented by $T = (t_1, t_2, \dots, t_m)$.
$\mathcal{C}$	State space of a camera that consists of finite set of discretized poses of camera. Each pose of camera is given by its pan, tilt and zoom value.
$c_i \in \mathcal{C}$	State of camera i, which is given by the discretized pan, tilt and zoom value.

$C \in \mathcal{C}^n$	Joint state of n cameras represented by $C = (c_1, c_2, \dots, c_n)$.
$fov(c_i) \subset \mathcal{T}_l$	Subset of target locations lying within the field of view (fov) of camera i in its state c_i.
$fov(C) \subset \mathcal{T}_l$	Subset of target locations lying within the joint fov of all cameras in state C, i.e., $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$.
$\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$	State space of MDP which consists set of tuples of joint state of m targets and n active cameras.
$S = (T, C) \in \mathcal{S}$	A state of MDP that consists of joint state of m targets and n active cameras.
$\mathcal{A}_i$	Action space of camera i.
a_i	Action of camera i such that $a_i \in \mathcal{A}_i$.
A	Joint action of all active cameras.
$\mathcal{A}$	Set of joint actions of all active cameras.
$G(\mathcal{Q}, \mathcal{E})$	Coordination Graph with $\lvert\mathcal{Q}\rvert = n$ nodes and $\lvert\mathcal{E}\rvert$ edges.
$\Gamma(i)$	Set of neighboring nodes of i.
μ_{ij}	A message sent from node i to node j in CG.

4.3.1 States and Actions

The state $S \in \mathcal{S}$ of the MDP framework consists of joint states of m targets $T \in \mathcal{T}^m$ and n active cameras $C \in \mathcal{C}^n$ such that $\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$ represents the state space of MDP framework. $\mathcal{T}$ and $\mathcal{C}$ represents the state space of each target and active camera respectively. A state $S = (T, C)$ represents a tuple that consists of joint state of targets and joint state of cameras such that $T = (t_1, t_2, \dots, t_m)$ and $C = (c_1, c_2, \dots, c_n)$. A state of target k is a tuple $t_k = (t_{l_k}, t_{d_k}, t_{v_k}) \in \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$ that consists of target's location $t_{l_k} \in \mathcal{T}_l$, direction $t_{d_k} \in \mathcal{T}_d$ and speed $t_{v_k} \in \mathcal{T}_v$ such that the state space of a target is $\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$. The state space of active camera $\mathcal{C}$ is a finite set of discrete pan/tilt/zoom positions. The set of target locations that lies within the fov of camera i in state c_i is denoted by $fov(c_i) \in \mathcal{T}_l$.

Therefore the joint fov of all the cameras is given by $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$. The depth of fov of the active cameras is limited such that, the images of the targets detected within the fov of each active camera satisfy a pre-defined resolution. The targets' states and cameras' states are fully observable to the agents. The joint actions of the active camera is defined as $A = (a_1, a_2, \ldots, a_n)$ where $a_i \in \mathcal{A}_i$ is the action of camera i and $\mathcal{A}_i$ is the action space of camera i which is the set of PTZ commands to move the camera to the specified position. The joint action space is denoted as $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n$.

4.3.2 Transition Model T_f

The transition probability $P(S'|S, A)$ of the MDP can be factored into transition probabilities of individual active cameras and targets after repeatedly applying the conditional independence property as discussed in Section 3.3.2 in Chapter 3. That is,

$$\begin{aligned} P(S'|S,A) &= P(C'|C,A)\, P(T'|T) \\ &= \prod_{i=1}^{n} P(c_i'|c_i,a_i) \prod_{k=1}^{m} P(t_k'|t_k) \\ &= \begin{cases} \displaystyle\prod_{k=1}^{m} P(t_k'|t_k) & \text{if } P(c_i'|c_i,a_i) = 1 \text{ for } i = 1,\ldots,n, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.3}$$

The transition probability $P(t_k'|t_k)$ of target k is calculated using Gaussian distributions as discussed in Section 3.3.2.1 in Chapter 3.

4.3.3 Objective/Reward Function R

As mentioned previously, the goal of the surveillance system is to maximize the number of targets observed in active cameras with a guaranteed image resolution. This goal is

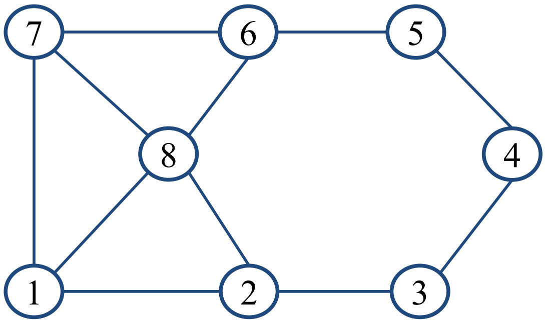


Figure 4.2: Coordination graph for the active camera network in Figure 4.1.

achieved by defining the reward function for pair of cameras $(i, j) \in \mathcal{E}$. We will see in the next section, that the global value function is decomposed into sum of pair of local value functions in the CG. Therefore, the reward function for a pair of cameras i and j given the states of all targets and states of camera i and j, is defined as follows:

$$R_{ij}((T, c_i, c_j)) \triangleq \sum_{k=1}^{m} \widetilde{R}_{ij}(t_k, c_i, c_j) \tag{4.4}$$

$$\widetilde{R}_{ij}(t_k, c_i, c_j) \triangleq \begin{cases} 1 & \text{if target } k\text{'s location lies in } fov(c_i) \text{ OR } fov(c_j), \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

where $\widetilde{R}_{ij}$ is the reward function of single target for a pair of cameras i and j. If the location of target k lies within $fov(c_i)$ or $fov(c_j)$, then the target k is guaranteed to be observed at a predefined image resolution, as discussed in Section 4.3.1, and $\widetilde{R}_{ij}(t_k, c_i, c_j) = 1$ results.

4.3.4 Coordination Graph Concepts

In multi-agent systems, the structure of the agents is often specified by the concept of Coordination Graph (CG) [Guestrin *et al.*, 2002]. The CG is an undirected graph represented as $G = (\mathcal{Q}, \mathcal{E})$ where each node $i \in \mathcal{Q}$ represents the agent that controls the i^{th} active camera and an edge $(i, j) \in \mathcal{E}$ represents the dependency between the nodes i and j. An agent i coordinates its actions only with the set of neighboring agents $\Gamma(i)$ that are connected to agent i in the CG. In active camera networks, the CG is constructed based on the proximity of fov of the cameras, i.e., (a) when the fov of two active cameras i and j are overlapping and (b) there is a non-zero transition probability for a target to transit between the fov of cameras i and j, then there can be an edge between the cameras i and j in the CG. The CG for the active camera network in Figure 4.1 is shown in Figure 4.2. In this CG, there is no edge between the cameras 2 and 6 because their fov is non-overlapping and it is not possible for a target to leave fov of camera 2 and enter immediately into the fov of camera 6 and vice-versa.

As mentioned previously, we use the CG to decompose the global coordination problem into number of local coordination problems each involving fewer cameras. In general, the global function $V(S, A)$ can be decomposed into sum of pair-wise local value functions. In our surveillance problem of maximizing the expected number of targets in active cameras, we approximate the decomposition of value function to gain the computational efficiency as follows:

$$V(S, A) \approx \sum_{(i,j)\in\mathcal{E}} V_{ij}((T, c_i, c_j), a_i, a_j) \tag{4.6}$$

where a_i and a_j are the actions of cameras i and j respectively, $V_{ij}((T, c_i, c_j), a_i, a_j)$ [1] is

[1] $V_{ij}((T, c_i, c_j), a_i, a_j)$ is a symmetric function.

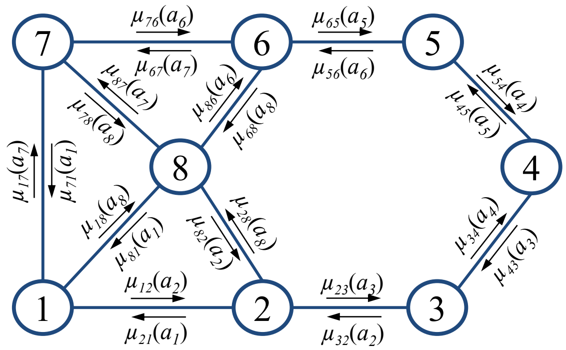


Figure 4.3: Message passing mechanism in max-plus algorithm for the active camera network in Figure 4.1.

the decomposed local value function for the pair of cameras i and j which is given by

$$V_{ij}((T, c_i, c_j), a_i, a_j) \quad = \sum_{T' \in \mathcal{T}^m} R_{ij}(T', c'_i, c'_j) P(T'|T). \tag{4.7}$$

The local value function in (4.7) is the expected number of targets observed in cameras i and j, and incurs computational time in the order of $\mathcal{O}(|\mathcal{T}|^m)$ which is exponential in number of targets. As discussed in Chapter 3, by exploiting the conditional independence property in the transition model, we can simplify (4.7) as sum of expected observation of each targets as shown below:

$$V_{ij}((T, c_i, c_j), a_i, a_j) \quad = \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} \widetilde{R}_{ij}(t'_k, c'_i, c'_j) P(t'_k|t_k). \tag{4.8}$$

The computational time in (4.8) incurs $\mathcal{O}(|\mathcal{T}|m)$ which is linear in number m of targets. The proof of (4.8) is given in Appendix A.2. It is assumed that the states of the targets are fully observable to all the agents, and also any agent i knows the state of its own camera and its neighboring cameras.

4.3.5 Max-plus algorithm

The optimal action a_i^* for each camera i can be estimated from the CG using the max-plus algorithm [Vlassis *et al.*, 2004; Kok and Vlassis, 2006]. In max-plus algorithm, each agent i repeatedly sends the message $\mu_{ij}(a_j)$ to its neighboring agents $j \in \Gamma(i)$ (see Algorithm 1). For a given state of targets T and cameras C, a message $\mu_{ij}(a_j)$ in max-plus algorithm is considered as a locally optimized value function between the agents i and j and is given by

$$\mu_{ij}(a_j) = \max_{a_i \in \mathcal{A}_i} \left\{ V_{ij}((T, c_i, c_j), a_i, a_j) + \sum_{k \in \Gamma(i) \setminus j} \mu_{ki}(a_i) \right\} - c_{ij} \tag{4.9}$$

```
Input: State of targets $T = (t_1, t_2, \ldots, t_m)$.
Output: Optimal actions of cameras $A^* = (a'_1, a'_2, \ldots, a'_n)$.
/* Initialize the messages and variables                         */
$\mu_{ij} = \mu_{ji} = 0$ for $(i, j) \in \mathcal{E}$, $iterCount = 0$;
while (fixed-point = false $\vee$ $iterCount < \kappa$) do
    /* run one iteration                                         */
    fixed-point=true, $iterCount$++;
    for every agent $i$ do
        for all neighbors $j = \Gamma(i)$ do
            Send $j$ message $\mu_{ij}(a_j)$ computed by (4.9)
            if $\mu_{ij}(a_j)$ differs from previous message by a small threshold then
                fixed-point=false
            end
            Determine $a'_i = \underset{a_i \in \mathcal{A}_i}{argmax} \sum_{j \in \Gamma(i)} \mu_{ji}(a_i)$
        end
    end
end
/* return the actions of all the agents                           */
return $A^* = (a'_1, a'_2, \ldots, a'_n)$ ;
```

Algorithm 1: Policy computation algorithm for MDP framework (MOMT) using max-plus algorithm.

where $\Gamma(i)\backslash j$ is a set of neighboring agents of i except j, $V_{ij}((T, c_i, c_j), a_i, a_j)$ is the local value function of cameras i and j, and $\sum\limits_{k\in\Gamma(i)\backslash j} \mu_{ki}(a_i)$ is the sum of messages that are received from the neighboring agents of i except j. The term $c_{ij} = \frac{1}{|\mathcal{A}_i|}\sum\limits_{a_i\in\mathcal{A}_i} \hat{\mu}_{ij}(a_j)$ is the normalization vector that prevents the explosion of values of the messages when there is a cycle in the graph and $\hat{\mu}_{ij}(a_j)$ is the unnormalized version of message $\mu_{ij}(a_j)$. Each message $\mu_{ij}(a_j)$ can be regarded as a local value function that maps an action a_j of camera j to a real number, i.e., $a_j \to \mu_{ij}(a_j)$. The messages are exchanged with their neighbors as shown in Figure 4.3, until they converge. That is, the values of the current messages does not vary too much from the previous messages. For tree structured graphs, the messages converge after a finite number of iterations [Pearl, 1988; Wainwright *et al.*, 2004]. For graphs with cycles, there is no theoretical guarantee for the messages to converge within a finite number of iterations and hence stop the algorithm after a finite κ iterations. However, the max-plus algorithm has been applied to many graphs with cycles in [Crick and Pfeffer, 2003; Kok and Vlassis, 2006]. At convergence, the optimal action for an agent i can be computed as follows:

$$a_i^* \quad = \quad \underset{a_i\in\mathcal{A}_i}{argmax} \sum_{j\in\Gamma(i)} \mu_{ji}(a_i) \tag{4.10}$$

and also the following holds [Pearl, 1988; Wainwright *et al.*, 2004]:

$$\sum_{j\in\Gamma(i)} \mu_{ji}(a_i) \quad = \quad \underset{\{A\in\mathcal{A}|a'_i=a_i\}}{max} \sum_{(i,j)\in\mathcal{E}} V_{ji}((T, c_i, c_j), a'_i, a_j). \tag{4.11}$$

Note that the optimal joint action computed by centralized MDP (4.2) may not be same as that of joint action $(a_1^*, a_2^*, \ldots, a_n^*)$ computed by (4.10). This is because (a) we decompose the global value function into sum of pair-wise local value functions and (b) max-plus algorithm computes the local optimal actions for each agent through exchanging messages until it converges or for a finite number of iterations. However, we have shown in experiments, that the performance of max-plus algorithm in active camera network is closer to the optimal solution given by (4.2) for different surveillance settings with $n = 6$ cameras.

4.4 Experiments and Discussion

This section evaluates our proposed work (denoted as MP in Figure 4.4, 4.7 and 4.8) to show that (a) its performance is close to the centralized MDP approach for CG with cycles for $n = 6$ cameras and (b) also performs well for increasing number of cameras simulated up to $n = 50$ cameras. We have simulated active camera networks in realistic setup using the Player/Stage simulator [Gerkey *et al.*, 2003] as described in Section 3.5.1 in Chapter 3. Our proposed work is empirically compared with some of the existing baseline camera control approaches and with our previous work in Chapter 3[2] as follows:

- *Centralized MDP framework* (MDP): The active cameras are controlled by the centralized MDP framework described in Chapter 3. We use this approach to evaluate the performance of our work when there are cycles in the CG.
- *Systematic approach* (Sys): The active cameras are panned systematically to each of its states in a round robin fashion for every time step; and
- *Static approach* ($Stat$): The active cameras are placed at a particular state such that they observe the maximum area of the environment.

We use the same performance metric $PercentObs$ as in Chapter 3 to evaluate the above approaches. It is given by,

$$PercentObs = \frac{100}{\tau M_{tot}} \sum_{i=1}^{\tau} M_{obs}^{i}$$

[2] We were able compute the solutions for centralized MDP (denoted as MDP in Figures 4.7 and 4.8) up to $n = 6$ cameras each having $|\mathcal{C}| = |\mathcal{A}_i| = 8$ states and actions.

where τ (i.e., set to 50 in simulations) is the total number of time steps taken in our experiments, M^i_{obs} is the total number of targets observed by the active cameras at a given time step i, and M_{tot} is the total number of targets present in the environment. That is, the $PercentObs$ metric averages the percentage of targets being observed by the active cameras over the entire duration of τ time steps. We will discuss the experimental setups followed by detailed analysis of experimental results.

4.4.1 Simulation Experiments: Setup

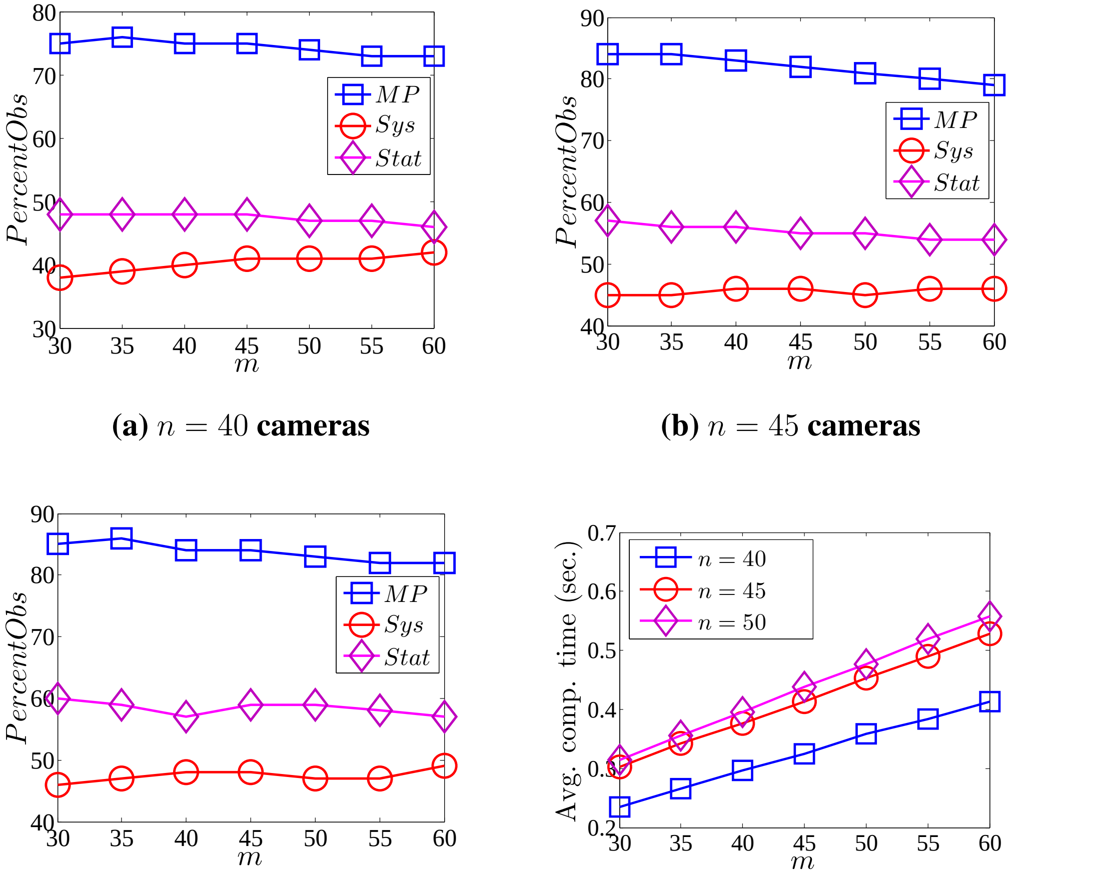
The proposed approach is evaluated for corridor, hall setup and a more complex large environment. The size of the corridor, hall and the complex setups are 40×7, 20×10 and 40×60 target locations respectively. We have tested up to $n = 6$ cameras and $m = 25$ targets in the corridor and hall setups, and also $n = 50$ cameras and $m = 60$ targets in the complex setup. The CGs for these setups are constructed manually based on the fov of cameras and their proximities as discussed in Section 4.3.4. The corridor and hall setups are used to evaluate the performance of our approach when the CG of the active cameras has varying number of cycles. This is done by adding additional edges to CGs for the hall and corridor setups. Whereas, the complex setup is used to evaluate performance when the number of cameras increases. Since we were able to compute the centralized MDP solution in real-time for only up to $n = 6$ cameras, the size of the hall and corridor setups are relatively small, and the number of cameras is relatively less when compared to the complex setup. We empirically determined that the algorithm converges within $\kappa = 10$ iterations for all most all the setups in our experiments. When they don't converge due to cycles in the graphs, we stop the algorithm after $\kappa = 10$ iterations. The active cameras and targets are simulated in the these environments as described in Section 3.5.1 in Chapter 3. The simulator itself acts as static cameras and provides the targets' location information.

4.4.2 Simulation Experiments: Results

Figure 4.4 (a) - (c) shows the performance of our approach for large complex setup with $n = 40$, 45, and 50 cameras. The simulation environment with $n = 50$ cameras is shown in Figure 4.5. We have simulated up to $m = 60$ targets in this environment and used the same set of target trajectories for surveillance settings with $n = 40$, 45, and 50 cameras. We have compared the performance of our approach with baseline Sys and $Stat$ approaches. The observations from this experiments are as follows:

In general, the percentage of targets observed is much better than the baseline approaches. Although the MP approach is an approximate solution, the average percentage of observation of targets over the results in Figure 4.4 (a)-(c) is around 80%. The Sys and $Stat$ approaches performs poorly because they are controlled independent of the targets' information. When we increase the number of cameras from $n = 40$ to 50, the performance of MP approach improves gradually as shown in Figure 4.4 (a)-(c). This is because, when new cameras are added to the same surveillance environment, the cameras overlaps each other and increases the probability of observing more number of targets. Hence the performance improves with increase in cameras. At the same time, when new cameras are added, the edges and the cycles in respective CG increases. This in turn increases the number of messages computed in each iteration and hence there is an increase in the average computation time of actions as shown in Figure 4.4(d).

We have also tested the robustness of our approach when there is a failure of cameras due to hardware malfunctioning. Figure 4.6 shows the performance of our approach when the number of cameras that fails (denoted as n_f in the graph) ranges from $n_f = 0$ to 5. It can be seen that, the performance of our approach is better than other approaches even when there is failure of few cameras. This is because, when a camera fails to send/receive messages from its neighboring cameras, then the remaining cameras can still coordinate with their neighbors to achieve the surveillance task. Failure of a camera will not affect the



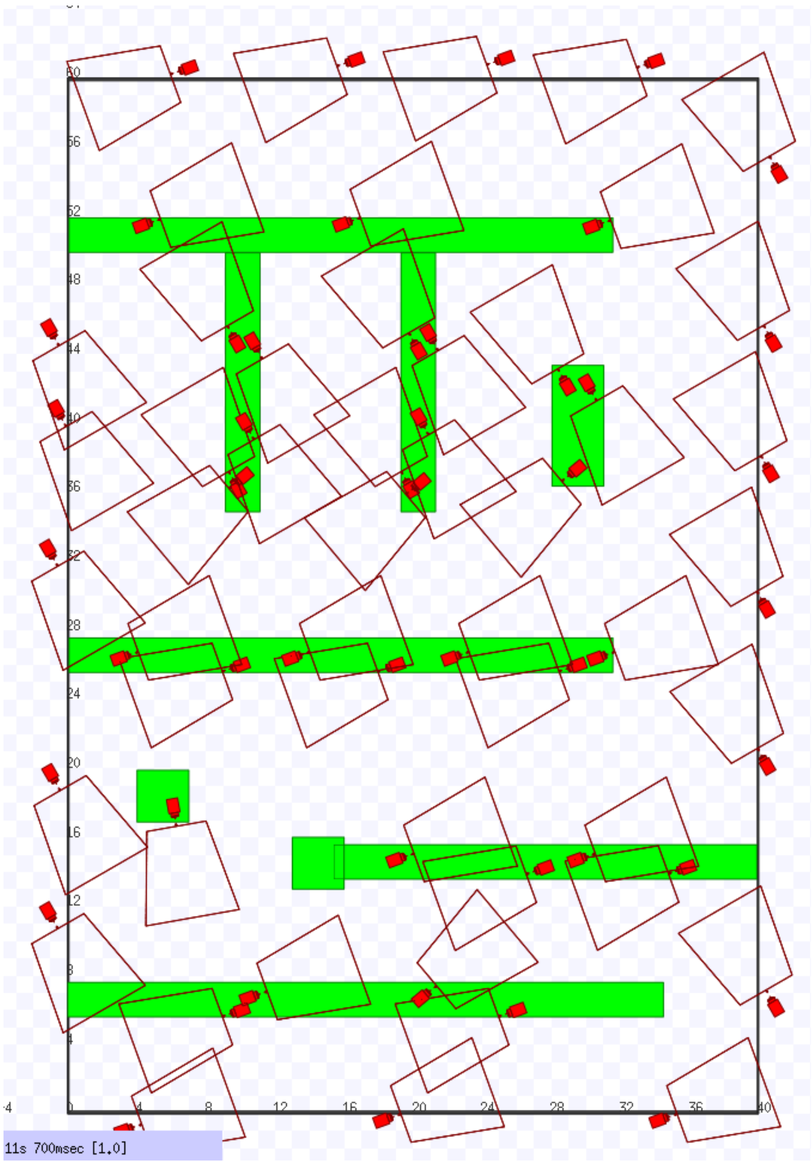
(a) $n = 40$ **cameras**

(b) $n = 45$ **cameras**

(c) $n = 50$ **cameras**

(d) Avg. comp. time vs Number m **of targets**

Figure 4.4: Comparison of $PercentObs$ vs. Number m of targets in large setup ($|\mathcal{T}_l|$=40×60 target locations) for (a) $n = 50$, (b) $n = 45$ and (c) $n = 40$ cameras. (d) Comparison of average computation time for MP approach in complex setup ($|\mathcal{T}_l|$=40×60 target locations) for $n = 40$, 45, and 50 cameras.



11s 700msec [1.0]

Figure 4.5: Complex camera setup ($|\mathcal{T}_l|$=40×60 target locations) simulated in Player/Stage simulator.

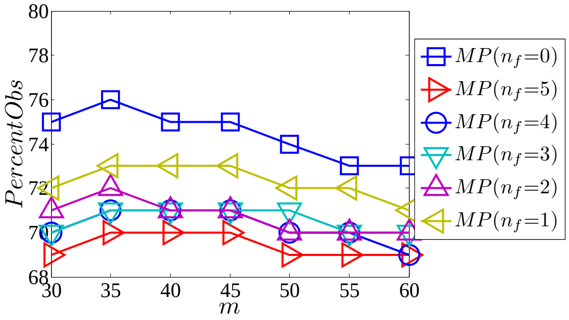


Figure 4.6: Robustness of max-plus algorithm for complex setup with $n = 40$ cameras and number of failure cameras $n_f = 1, 2, 3, 4$ and 5.

coordination of entire camera network, instead only the performance of the surveillance degrades gracefully as shown in the Figure 4.6.

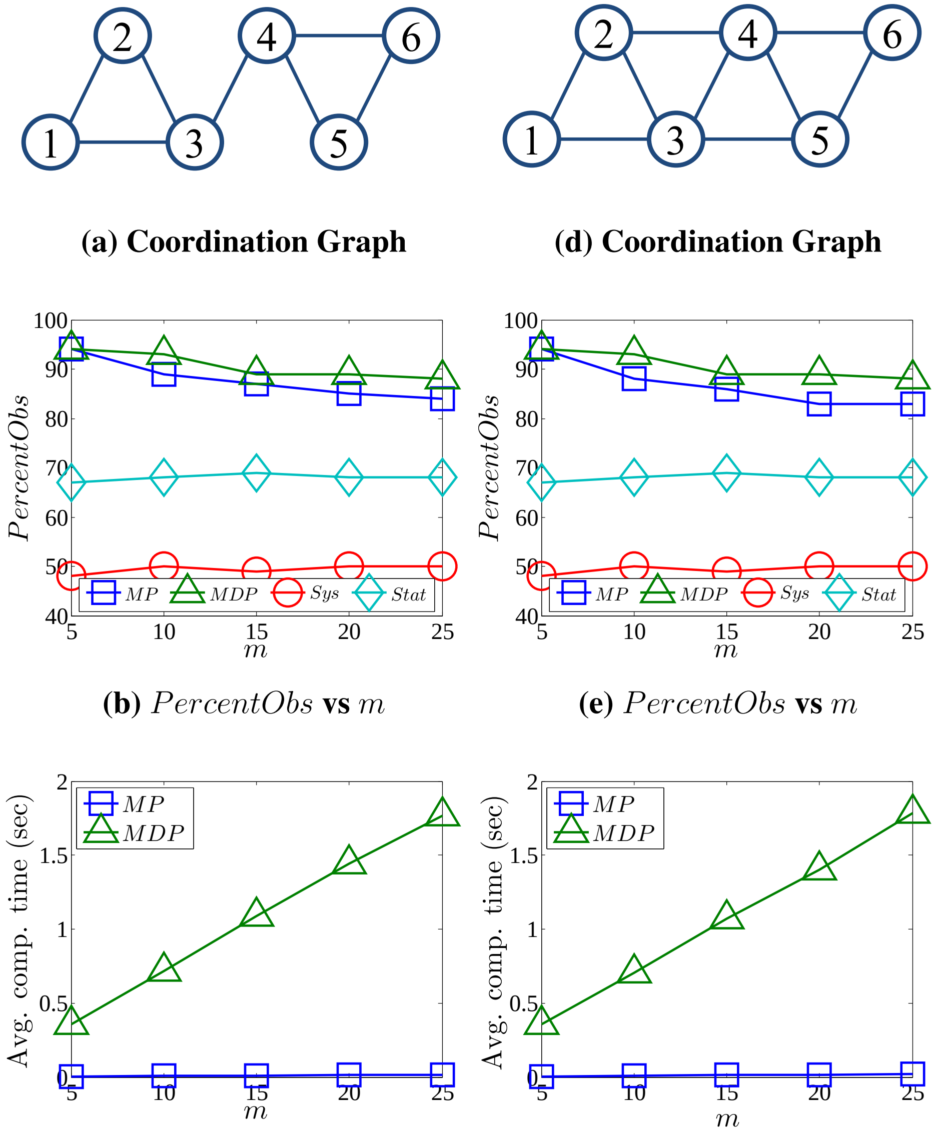
Figure 4.7 and 4.8 shows the performance of our MP approach compared with MDP, Sys and $Stat$ approaches in corridor and hall setups, respectively. In both these setups, we have positioned the cameras such that the cameras overlap and have more cycles in the CGs. The CGs for corridor setup in Figure 4.7(a) and (d) has 7 and 9 edges, respectively, and the CGs for hall setup in Figure 4.8(a) and (d) has 8 and 10 edges, respectively. The observations from this experiments are as follows:

When there are cycles in the CG, the performance of our approach is close to the MDP approach and much better than the Sys and $Stat$ approaches. This is because the Sys and $Stat$ approaches are controlled independent of the targets' information. It is expected that the performance of MP approach to be slightly less than the MDP approach because, MP approach is an approximate solution (see Section 4.3.5), whereas MDP approach is the centralized solution. It is observed that the performance of MP approach is better, when there are less cycles in the CG (see Figure 4.7(b) and 4.8(b)). When the cycles in the CG is increased by adding additional edges, the performance of MP approach drops gradually (see Figure 4.7(e) and 4.8(e)). In practice, there is less chance of having many cycles in CGs for active camera networks (as in Figure 4.7(d) and 4.8(d)) because, it is less likely that many active cameras will be placed close to each others. It is also observed that the $Stat$ approach performs better than Sys approach because, the cameras are placed close to each other in order to have cycles in the CG. In such settings, when the cameras are fixed at a position where they can cover maximum area, then almost half of the total surveillance area is covered by the 6 cameras. Hence $Stat$ approach observes more number of targets than Sys approach.

We have also compared the average computation time incurred to compute the actions in MP and MDP approach (see Figure 4.7(c), (f) and 4.8(c), (f)). When the number of targets increases, the computation time for MDP approach increases at a higher rate than MP approach. This is because, the expected number of targets is calculated for all possible joint action space in MDP approach. Whereas in MP approach, the expected number of targets is calculated only for the action space of each individual cameras separately.

Below we summarize the observations from our experiments:

- The message passing mechanism and the decomposed value function (4.8), helps to compute the actions of individual cameras, which makes our MP approach scalable for increasing number of cameras and targets.
- The performance of our MP approach is much better than baseline approaches even for $n = 50$ cameras and robust for camera or communication failures.
- Although the MP approach is an approximate solution, its average performance is about 80% for our large setup with $n = 40, 45$ and 50 cameras.
- Performance of our approach is closer to the centralized MDP solution and much better than Sys and $Stat$ approaches for $n = 6$ cameras and CGs with cycles. When the number of targets increases, the average computation time of centralized MDP increases at a higher rate than our approach due to exponential size of in joint action space.



(a) Coordination Graph

(d) Coordination Graph

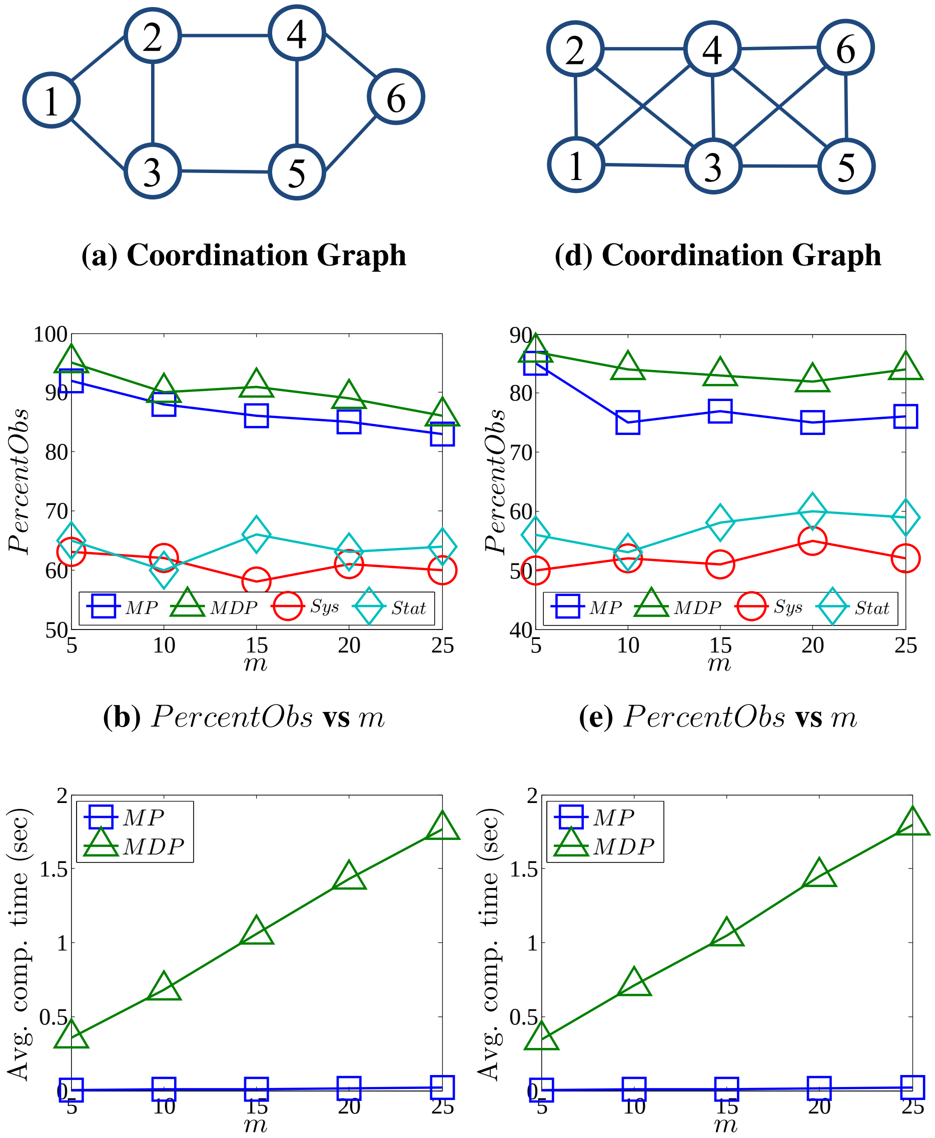
(b) $PercentObs$ **vs** m

(e) $PercentObs$ **vs** m

(c) Avg. comp. time vs m

(f) Avg. comp. time vs m

Figure 4.7: Comparison of $PercentObs$ and average computation time with number m of targets for corridor setup ($|\mathcal{T}_l|$=40×7 target locations) whose CG has 7 edges in (a) - (c) and 9 edges in (d) - (f).



(a) Coordination Graph

(d) Coordination Graph

(b) $PercentObs$ **vs** m

(e) $PercentObs$ **vs** m

(c) Avg. comp. time vs m

(f) Avg. comp. time vs m

Figure 4.8: Comparison of $PercentObs$ and average computation time with number m of targets for hall setup ($|\mathcal{T}_l|$=20×10 target locations) whose CG has 8 edges in (a) - (c) and 10 edges in (d) - (f).

4.5 Summary

In this chapter, we have extended the centralized MDP framework in Chapter 3 to improve the scalability in number of cameras. We have used the concept of coordination graph to decompose the centralized coordination problem into sum of local coordination problems for each pair of active cameras. Then we have used the max-plus algorithm to solve the coordination problem approximately through message passing across the edges of the coordination graph. In this work, the proposed camera coordination framework is scalable in number of cameras because each camera repeatedly exchanges the messages with their neighbors in the coordination graph and computes their individual optimal action over the received messages. Since we extend our previous work in Chapter 3, we exploit the same conditional independence properties in transition model to decompose the value function for pair of cameras to improve the scalability in number of targets to be observed. We have empirically evaluated the proposed approach for different surveillance settings with cycles in the CGs. The experimental results show that the our proposed work can scale up to 50 cameras and 60 targets in real-time.

Chapter 5

FOMT Problem in Fully Observable Surveillance Environment

5.1 Introduction

This chapter presents a novel decision-theoretic approach to control and coordinate active cameras for achieving the **fairness in observation of multiple targets (FOMT)** under *fully observable* surveillance environment. In Chapter 3 and 4, the active cameras are controlled to maximize the number of targets observed with guaranteed resolution. But there is no notion of fairness property in observing the targets in active cameras.

Fairness in active camera surveillance is an important property in which the active cameras (resources) are scheduled or allocated to observe the targets (jobs) in the environment such that no target is "starved" of observation by the cameras for long time. This property is important in many real-world surveillance of industrial sites, airport terminals, train stations, schools and university campuses, etc. The lack of fairness in active camera surveillance may lead to situations where few targets may not be observed for long duration of time. When targets are not observed for long duration, their activities will not be cap-

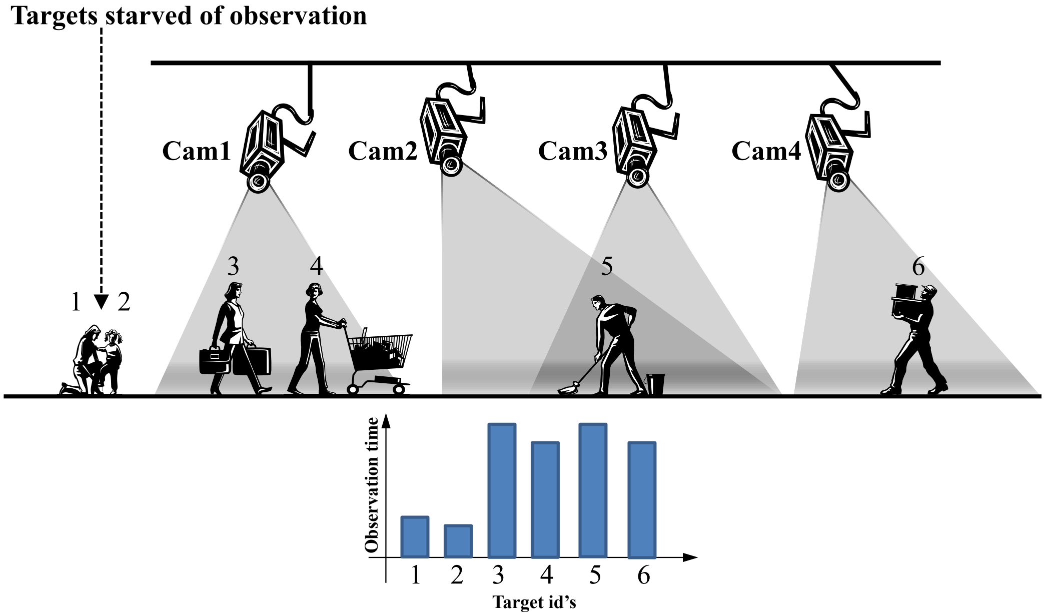


Figure 5.1: Example scenario of targets starved of observation by active cameras and their respective observation times.

tured/monitored in the active cameras. This leads to a loophole in surveillance, where the suspicious or anti-social behavior of the targets will neither be captured in the active cameras for biometric tasks, nor recorded in videos for future forensic investigations.

Existing multi-camera coordination algorithms lacks this property of fairness in observation of multiple targets (see Table 2.2) and hence "starve" some targets of observation by the active cameras for a prolonged period of time (e.g., see Figure 5.1), especially those isolated ones with low likelihood of observing them. In the worst case, those targets may not be observed at all. Surprisingly, this issue of starvation has not been tackled by the multi-camera surveillance community. It motivates the need to design and develop a coordination framework that can coordinate the actions of the active cameras to observe all targets fairly. Intuitively, this implies prioritizing the observation of targets with the least observation time such that a fair observation of all the targets is achieved if and only if increasing the observation of any target necessarily results in a decrease in observation of some other target with equal or lower observation time. Such a notion of fairness is often known as the max-min fairness in resource allocation problems (e.g., bandwidth allocation in networking).

Achieving fair observation of targets in active camera surveillance is challenging and non-trivial because: (a) real-world physical constraints such as the spatial localities of the active cameras (i.e., resources) and moving targets (i.e., users) restrict their interactions. For example, some cameras may not be able to observe any target at times because all the targets are beyond their possible fields of view (fov's). On the other hand, some targets may occasionally move into regions that are occluded from observations by the active cameras; (b) the stochastic motion of the targets entail uncertain (hence, less predictable) trajectories, which complicate how the active cameras are to be coordinated to keep possibly multiple targets of the least observation time within their fov's at a guaranteed predefined image resolution; and (c) the proposed coordination framework, if poorly designed, incurs exponential time in the number of targets to be observed during surveillance, thus degrading its real-time performance. Therefore it is necessary to address these issues in the coordination framework in order to preserve fairness in observation of targets.

In this chapter, we propose a novel, principled decision-theoretic approach to control and coordinate active cameras to achieve max-min fairness under uncertainty in the observation of multiple moving targets (Section 2.2). Our decision-theoretic approach is based on Markov Decision Process (MDP) framework (Section 5.3) that addresses the above-mentioned issues in the following ways: (a) the notion of fairness in multi-target observation can be formally realized in the domain of multi-camera surveillance for the first time by exploiting the max-min fairness metric (Section 5.4.3) to formalize our surveillance objective, that is, to maximize the expected minimum observation time over all targets while guaranteeing a predefined image resolution of observing them (Section 5.5); (b) the uncertainty in the motion and observation times of the targets can be modeled probabilistically (Section 5.4.2); and (c) the structural properties and assumptions of a surveillance environment can be exploited to improve the scalability of our coordination framework to linear time in the number of targets to be observed during surveillance (Section 5.5). Our proposed MDP framework is empirically evaluated in various realistic surveillance environmental setups through extensive simulations and real Axis 214 PTZ cameras (Section 5.6).

5.2 Background

In this section, we will study some of the fairness metrics that are used in the networking and scheduling literatures. Jain's fairness index [Jain *et al.*, 1984] is a popular metric used to measure whether the users or applications receive fair share of the system bandwidth and the resources. The value of the Jain's index lies between 0 and 1, such that the system is perfectly fair, if the index is 1 and the system is unfair, if the index is 0. Another popular fairness metric that is used in networking systems is the max-min index [Tassiulas and Sarkar, 2002], which maximizes the minimum share by a user, where the resources are allocated based on the order of increasing demand. This index evenly divides the resources among the users such that a user does not receive more resources if it results in reducing the resources of the least serviced user. Proportional fairness tries to maximize the total network throughput, while at the same time allowing all users to access at least a minimal level of resources. This index assigns each user a weight that is inversely proportional to the anticipated resource consumption of that user [Liu and Leung, 2008]. Proportional fairness is equivalent to weighted fairness queuing [Lee *et al.*, 2007] when the weight of i^{th} channel is inverse of cost per data bit of data flow in that channel. Resource Allocation Queueing Fairness Measure [Raz *et al.*, 2004] is used in allocating resources in queues, where at every point in time, every job in the queue deserves an equal service rate. Fairness of the policy is determined based on the variations of the service rate of the jobs.

However, fairness in surveillance is different from the networking and scheduling domains. Our fairness problem in surveillance is more general than a typical resource allocation problem, i.e., when some resource is allocated to a certain job, it is assumed that, this same resource cannot be exploited by other jobs. But whereas in surveillance, when a camera (i.e., resource) is allocated to observe certain target (i.e., job), the same camera can be exploited to observe other targets that are residing in the fov of that camera.

5.3 System Overview

The proposed surveillance framework consists of a supervised surveillance environment and a MDP camera controller, similar to the setup in Chapter 3. The surveillance environment consists of targets, static and active cameras. The targets are moving objects whose motion is stochastic in nature. The static cameras are wide-view cameras and observe the surveillance environment only at low-resolution. The PTZ cameras are controlled by MDP controller to achieve max-min fairness in observing the targets. Formally, the proposed MDP framework is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}_f, \mathcal{R})$ consisting of:

- a set $\mathcal{S}$ of joint states of active cameras and targets in the surveillance environment (Section 5.4.1);
- a set $\mathcal{A}$ of joint actions of active cameras (Section 5.4.1);
- a transition function $\mathcal{T}_f : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denoting the probability $P(S'|S, A)$ of going from the current joint state $S \in \mathcal{S}$ to the next joint state $S' \in \mathcal{S}$ using the joint action $A \in \mathcal{A}$ (Section 5.4.2); and
- a real-valued objective/reward function $R : \mathcal{S} \rightarrow \mathbb{R}$ representing a high-level surveillance goal (Section 5.4.3).

A policy function π in MDP maps every state S to a joint action of cameras A, i.e., $\pi : S \rightarrow A$. We need to compute optimal policy π^* that maximizes the expected reward for a given state. This is computed by a greedy algorithm as described in Section 5.5. At every time step of surveillance, the MDP controller computes an optimal policy (i.e., optimal action) based on the observed state of the surveillance environment (from the static cameras), such that the max-min fairness is achieved. When the number of targets increases, the state space grows exponentially and hence the policy computation incurs exponential computation time. But, in our current work, we exploit the structure of our surveillance problem to reduce the exponential computation into linear time. Its important to mention the reader that the trade-off between the fairness and coverage of the system has not been considered in this work at this point of time. Therefore, in our current work, the cameras are controlled only to improve the fairness property in the active camera surveillance and not the coverage.

5.4 Problem Formulation

5.4.1 States and Actions

A joint state $S \in \mathcal{S}$ of our MDP controller is defined as a pair of joint states $T_{\mathcal{M}} \in \mathcal{T}^m$ of m targets and $C \in \mathcal{C}^n$ of n active cameras where $\mathcal{T}$ and $\mathcal{C}$ denote sets of all possible states of each target and active camera, respectively and $\mathcal{M} = \{k \in \mathbb{N} : 1 \leq k \leq m \mid$ 'k' is an index of a target$\}$. That is, $S \triangleq (T_{\mathcal{M}}, C)$ and $\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$. Let $T_{\mathcal{M}} \triangleq (t_1, t_2, \ldots, t_m) \in \mathcal{T}^m$ and $C \triangleq (c_1, c_2, \ldots, c_n) \in \mathcal{C}^n$ where $t_k \in \mathcal{T}$ and $c_i \in \mathcal{C}$ denote the corresponding states of target k and camera i, and $\mathcal{M} = \{1, 2, \ldots, m\}$. Let $t_k \triangleq (t_{l_k}, t_{d_k}, t_{v_k}, t_{o_k}) \in \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v \times \mathcal{T}_o$ where $t_{l_k}, t_{d_k}, t_{v_k}$, and t_{o_k} denote target k's location, direction, speed, and observation time, respectively. That is, $\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v \times \mathcal{T}_o$. The state space $\mathcal{C}$ of an active camera is a finite set of discrete pan/tilt/zoom positions. Let $fov(c_i) \subset \mathcal{T}_l$ be a subset of target locations lying within the fov of camera i in its state c_i. The joint fov of all cameras in joint state C is defined as $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$. The depth of fov of each active camera is limited such that imageries of the targets detected within its fov satisfy a pre-defined resolution. This is done by adjusting the zoom parameter of each camera based on its position. The joint actions of the camera controller are PTZ commands that move the corresponding cameras to their specified states. Let a joint action of the n cameras be denoted by $A \triangleq (a_1, a_2, \ldots, a_n) \in \mathcal{A}$ where a_i denotes the PTZ command of camera i. It is worth to note that in Chapters 3, 4 and 6, the state of a target comprise of its location, direction and speed. In this chapter, the observation time of a target is included in its state, in addition to its location, direction and speed. The notations and symbols used in this chapter are summarized in Table 5.1.

Table 5.1: Mathematical notations and its descriptions used in Chapter 5.

Notation	Description
n	Number of active cameras.
m	Number of targets to be monitored by n active cameras, such that $n \ll m$.
$\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v \times \mathcal{T}_o$	State space of a target represented by a set of tuples of location, direction, speed and observation time.
$\mathcal{T}_l$	Set of all possible discretized locations of the target in the surveillance environment.
$\mathcal{T}_d$	Set of all possible discretized directions of target.
$\mathcal{T}_v$	Set of discretized speeds of the target.
$\mathcal{T}_o$	Set of discretized observation time of the target.
t_k=$(t_{l_k}, t_{d_k}, t_{v_k}, t_{o_k})$	State of a target k is a tuple consisting of its location $t_{l_k} \in \mathcal{T}_l$, direction $t_{d_k} \in \mathcal{T}_d$, speed $t_{v_k} \in \mathcal{T}_v$ and observation time $t_{o_k} \in \mathcal{T}_o$.
$\mathcal{M}$	Set of indices of targets.
$T_{\mathcal{M}} \in \mathcal{T}^m$	Joint state of m targets represented by $T_{\mathcal{M}} = (t_1, t_2, \ldots, t_m)$ where $\mathcal{M} = \{1, 2, \ldots, m\}$.
$\mathcal{C}$	State space of a camera that consists of finite set of discretized poses of camera. Each pose of camera is given by its pan, tilt and zoom value.
$c_i \in \mathcal{C}$	State of camera i, which is given by the discretized pan, tilt and zoom value.
$C \in \mathcal{C}^n$	Joint state of n cameras represented by $C = (c_1, c_2, \ldots, c_n)$.

$fov(c_i) \subset \mathcal{T}_l$	Subset of target locations lying within the field of view (fov) of camera i in its state c_i.
$fov(C) \subset \mathcal{T}_l$	Subset of target locations lying within the joint fov of all cameras in state C, i.e. $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$.
$\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$	State space of MDP that consists of set of tuples of joint state of m targets and n active cameras.
$S = (T_{\mathcal{M}}, C) \in \mathcal{S}$	A state of MDP that consists of joint state of m targets and n active cameras.
a_i	Action of camera i is a PTZ command to move the camera to the specified state.
A	Joint action of all active cameras represented by a tuple $A = (a_1, a_2, \ldots, a_n)$.
$\mathcal{A}$	Set of joint actions of all active cameras.
$\mathcal{Y}$	Set of indices of targets whose observation time is minimum of all other targets such that $y = \lvert\mathcal{Y}\rvert$, $\mathcal{Y} \subseteq \mathcal{M}$ and $\forall i \in \mathcal{Y} : t_{o_i} = \underset{j=1\ldots m}{min}(t_{o_j})$.
$T_{\mathcal{K}}$	Joint state of k targets whose indices are in the set $\mathcal{K}$, i.e., $T_{\mathcal{K}} = (t_i)_{i \in \mathcal{K}}$ and $T_{\mathcal{K}} \in \mathcal{T}^k$ where $\mathcal{T}^k$ is a set of joint states of k targets.
$T_{\overline{\mathcal{K}}}$	Joint state of $(m-k)$ targets whose indices are in $\overline{\mathcal{K}} = (\mathcal{M} \backslash \mathcal{K})$, i.e., $T_{\overline{\mathcal{K}}} = (t_i)_{i \in \overline{\mathcal{K}}}$ such that $T_{\overline{\mathcal{K}}} \in \mathcal{T}^{(m-k)}$ where $\mathcal{T}^{(m-k)}$ is a set of joint states of $(m-k)$ targets.
$\mathcal{T}_C$	Set of states of a target whose locations lies in the $fov(C)$, i.e., $\forall t_i \in \mathcal{T}_C$ we have $t_{l_i} \in fov(C)$.
$\mathcal{T}_C^k$	Set of joint states of k targets whose locations lies in the $fov(C)$, i.e., $\forall T_{\mathcal{K}} \in \mathcal{T}_C^k$ we have $T_{\mathcal{K}} = (t_i)_{i \in \mathcal{K}}$ such that $\forall t_i = (t_{l_i}, t_{d_i}, t_{v_i}, t_{o_i})$ in $T_{\mathcal{K}}$ we have $t_{l_i} \in fov(C)$.
$(\mathcal{T}^k \backslash \mathcal{T}_C^k)$	Set of joint states of k targets where at least one of the target does not lies in the $fov(C)$, i.e., $\forall T_{\mathcal{K}} \in (\mathcal{T}^k \backslash \mathcal{T}_C^k)$ we have $T_{\mathcal{K}} = (t_i)_{i \in \mathcal{K}}$ such that $\exists t_i = (t_{l_i}, t_{d_i}, t_{v_i}, t_{o_i})$ in $T_{\mathcal{K}}$ such that $t_{l_i} \in \overline{fov(C)}$.

5.4.2 Transition Model T_f

By exploiting the following structural assumptions in the state transition dynamics of the surveillance environment:

- camera i's next state c'_i is conditionally independent of the other $n-1$ cameras' states and actions and the m targets' states given its current state c_i and action a_i for $i = 1, \ldots, n$ and
- target k's next state t'_k is conditionally independent of other $m-1$ targets' states (i.e., every target moves independently) given its current state t_k for $k = 1, \ldots, m$ and the cameras' next joint state C',

the transition model T_f can be factored into transition models of individual targets and active cameras, hence significantly reducing the time incurred to compute the optimal policy π^* for a given state S (Section 5.5). Furthermore, since the modern active cameras are able to move to their specified positions accurately [Axi, 2011], it is practical to assume the transition model of each individual camera to be deterministic and consequently represented by a function τ that moves camera i from its current state c_i to its next state $\tau(c_i, a_i)$ by the action a_i. Then, the transition model of the camera controller can be simplified to

$$\begin{aligned} P(S'|S,A) &= P(T'|T,C')P(C'|C,A) \\ &= \prod_{k=1}^{m} P(t'_k|t_k,C')P(C'|C,A) \\ &= \begin{cases} \displaystyle\prod_{k=1}^{m} P(t'_k|t_k,C') & \text{if } P(c'_i|c_i,a_i) = 1 \text{ for } i = 1,\ldots,n, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{5.1}$$

Derivation of (5.1) is followed from Section 3.3.2 in Chapter 3. The state transition of target k from t_k to t'_k includes stochastic transitions of its location from t_{l_k} to t'_{l_k}, its direction from t_{d_k} to t'_{d_k}, its speed from t_{v_k} to t'_{v_k}, and its observation time from t_{o_k} to t'_{o_k}. So, the transition probability of target k can be factored into transition probabilities of its location, direction, speed and observation time:

$$P(t'_k|t_k, C') = P(t'_{l_k}|t_{l_k}, t'_{d_k}, t'_{v_k})P(t'_{d_k}|t_{d_k})P(t'_{v_k}|t_{v_k})P(t'_{o_k}|t'_{l_k}, t_{o_k}, C') \ .$$

The transition probabilities $P(t'_{d_k}|t_{d_k})$ and $P(t'_{v_k}|t_{v_k})$ of the target's direction and speed are, respectively, modeled as Gaussian distributions $\mathcal{N}(\mu_d, \sigma_d)$ and $\mathcal{N}(\mu_v, \sigma_v)$ with the means μ_d and μ_v being the current direction and speed of the target, and σ_d and σ_v being the variance parameters which are learned from a dataset of the targets' trajectories in the environment. The transition probability $P(t'_{l_k}|t_{l_k}, t'_{d_k}, t'_{v_k})$ of the target's next location is constructed using the general direction-speed motion model, as described in Section 3.3.2.1 in Chapter 3. The observation time t_{o_k} of a target k is increased to $t'_{o_k} = (t_{o_k} + 1)$, if the target is observed in fov of any of the active cameras, else the observation time remains the same, i.e., $t'_{o_k} = t_{o_k}$. Therefore, the transition probability $P(t'_{o_k}|t'_{l_k}, t_{o_k}, C')$ is defined as follows,

$$P(t'_{o_k} = (t_{o_k} + 1)|t'_{l_k}, t_{o_k}, C') \quad = \quad \begin{cases} 1 & \text{if } t'_{l_k} \in fov(C'), \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

$$P(t'_{o_k} = t_{o_k}|t'_{l_k}, t_{o_k}, C') \quad = \quad \begin{cases} 1 & \text{if } t'_{l_k} \notin fov(C'), \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

5.4.3 Objective/Reward function R

Supposing the transition models of all targets are deterministic (i.e., the states of all targets in the next time step are known), the surveillance objective can be defined directly in terms of the max-min fairness metric, that is, to maximize the minimum observation time over all targets while guaranteeing a predefined image resolution of observing them. Such a surveillance objective can be achieved by defining an objective/reward function R that measures the minimum observation time over all targets:

$$R(S) = R((T_{\mathcal{M}}, C)) \triangleq \min_{k \in \mathcal{M}} t_{o_k} \ . \tag{5.4}$$

It is noteworthy to point out the usefulness of other popular fairness metrics to the domain of multi-camera surveillance such as the Jain's fairness index [Jain *et al.*, 1984]. Jain's fairness index measures whether the users receive their fair share of the resources. This index is revised to reflect our notations and given below:

$$\frac{(\sum_{k \in \mathcal{M}} t_{o_k})^2}{|\mathcal{M}| \sum_{k \in \mathcal{M}} t_{o_k}^2} \ .$$

The value of Jain's fairness index lies between 0 and 1. The observation of all targets is perfectly fair if the index is 1. It is unfair if the index is 0. Jain's fairness index is not suitable for measuring fairness in the observation of multiple targets in active multi-camera surveillance: For example, when 999 targets are always observed by any active camera and only 1 target is not being observed at all, Jain's fairness index yields 0.999, which is close to perfect. However, the single target is not observed at all and may potentially be a suspicious target that is critical to be observed by surveillance.

5.5 Policy Computation

As stated earlier, we need to compute the optimal policy π^* that maximizes the expected reward for a given state. In practice, the states of the targets in the next time step are uncertain due to stochasticity of their motion. Therefore, the optimal policy π^* has to instead maximize the expected minimum observation time over all targets in the next time step:

$$\pi^*((T_{\mathcal{M}}, C)) = A^* = \arg\max_{A \in \mathcal{A}} V(T_{\mathcal{M}}, C, A) \tag{5.5}$$

$$V(T_{\mathcal{M}}, C, A) \triangleq \sum_{T'_{\mathcal{M}} \in \mathcal{T}^m} R((T'_{\mathcal{M}}, C'))P(T'_{\mathcal{M}}|T_{\mathcal{M}}, C') \tag{5.6}$$

where $T'_{\mathcal{M}}$ and C' are, respectively, the joint states of the targets and active cameras in the next time step. The next joint state C' of the cameras can be determined deterministically from their current joint state C and joint action A using the function $c'_i = \tau(c_i, a_i)$ for $i = 1, \ldots, n$ (Section 5.4.2).

Computing an optimal policy π^* (5.5) incurs $\mathcal{O}(|\mathcal{A}||\mathcal{T}|^m)$ time that is exponential in number m of targets. This exponential time complexity can be significantly reduced by exploiting the structural property and assumptions of the surveillance environment, that is, the conditional independence property in the transition model (Section 5.4.2). As a result, the value function V (5.6) can be simplified to

$$V(T_{\mathcal{M}}, C, A) = t_{\text{min}} + \prod_{k \in \mathcal{Y}} \sum_{t'_k \in \mathcal{T}_{C'}} P(t'_k|t_k, C') \tag{5.7}$$

where $\mathcal{Y} \subseteq \mathcal{M}$ denotes the set of indices of all targets with minimum observation time in the current time step (i.e., $\mathcal{Y} \triangleq \{j \in \mathcal{M} \mid t_{o_j} = \min_{k \in \mathcal{M}} t_{o_k}\}$), $\mathcal{T}_{C'}$ denotes the set of a target's states whose locations are observed by the active cameras in their joint state C' (i.e., $\mathcal{T}_{C'} \triangleq \{t'_k \in \mathcal{T} \mid t'_{l_k} \in fov(C')\}$), and $t_{\text{min}} = \min_{k \in \mathcal{M}} t_{o_k}$ denotes a constant representing the minimum observation time over all targets in the current time step. The derivation of (5.7) is given in Appendix A.3. By plugging (5.7) into (5.6), (5.5) reduces to

$$\pi^*((T_{\mathcal{M}}, C)) = A^* = \arg\max_{A \in \mathcal{A}} \prod_{k \in \mathcal{Y}} \sum_{t'_k \in \mathcal{T}_{C'}} P(t'_k|t_k, C'). \tag{5.8}$$

The policy given by (5.8) chooses a joint action $A \in \mathcal{A}$ that maximizes the product of likelihoods of observing, in the next time step, all targets with minimum observation time in the current time step by active cameras in their joint state C'.

The following result indicates that an optimal joint action A^* (5.8) can be derived using linear time in the number m of targets to be observed during surveillance:

Theorem 3. *If (5.1) holds, then computing the optimal joint action A^* (5.8) incurs $\mathcal{O}(|\mathcal{A}||\mathcal{T}|m)$ time.*

In (5.8), computing the likelihood of observing a target with minimum observation time (i.e., sum of probabilities) incurs $\mathcal{O}(|\mathcal{T}|)$ time. Computing the product of $|\mathcal{Y}|$ likelihoods then incurs $\mathcal{O}(|\mathcal{T}|m)$ time since the size of $\mathcal{Y}$ can be m in the worst case and Theorem 3 follows.

As mentioned above, computing an optimal policy π^* (5.8) only needs to consider all targets with minimum observation time. These targets may be beyond the fov's of some active cameras due to their spatial localities, which is an issue stated in Section 5.1. Consequently, multiple possible optimal joint actions are possible because any action of such a camera is optimal. In the worst case, all targets with minimum observation time are beyond the fov's of all active cameras (i.e., $V(T_{\mathcal{M}}, C, A^*) = t_{\text{min}}$), thus causing all the cameras to be in "limbo".

To remedy this, the key idea is to repeatedly refine the set of optimal joint actions by preserving fairness in the observation of the remaining targets using (5.8) after ignoring those with minimum observation times. To elaborate, the first step is to compute the set $\mathcal{A}^*$ of optimal joint actions of the active cameras satisfying (5.8). Then, ignore the targets with minimum observation time by removing $\mathcal{Y}$ from $\mathcal{M}$, that is, $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{Y}$. Finally, consider $\mathcal{A}^*$ to be the new joint action space in (5.8), that is, $\mathcal{A} \leftarrow \mathcal{A}^*$. These steps are repeated until there is a unique optimal joint action A^* or the number $|\mathcal{M}|$ of remaining targets after ignoring those with minimum observation times is 0 (see Algorithm 2).

```
Input: State of targets $T = (t_1, t_2, \ldots, t_m)$.
Output: Optimal actions of cameras $A^* = (a'_1, a'_2, \ldots, a'_n)$.
while $(|\mathcal{A}| > 1 \vee |\mathcal{M}| > 0)$ do
    /* Compute the optimal joint action set                    */
    Compute $\mathcal{A}^* \triangleq \{A^*\}$ by (5.8)
    /* Eliminate targets whose indices are in $\mathcal{Y}$    */
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{Y}$
    /* Update the joint action set                             */
    $\mathcal{A} \leftarrow \mathcal{A}^*$
end
/* Optimal action for improving fairness                       */
return $A^* = (a'_1, a'_2, \ldots, a'_n)$
```

Algorithm 2: Policy computation algorithm of MDP framework to achieve FOMT.

5.6 Experiments and Discussion

This section empirically evaluates the performance of our proposed MDP framework in different realistic surveillance environmental setups in simulation and also in real Axis 214 PTZ cameras to demonstrate its feasibility in real-world surveillance. Interested readers can view our demo video here[1]. We have used Player/Stage simulator [Gerkey *et al.*, 2003] to evaluate the proposed approach in realistic surveillance setups. As stated in Chapter 2, the existing works on control and coordination of active cameras either focus & track one or more targets or try to maximize the number of targets. Hence it is fair to compare and contrast our framework with the existing fairness metrics and with baseline camera control approaches. Our MDP approach that is based on max-min fairness metric (denoted as MM in Figure 5.3 and 5.4) is compared against the following existing fairness metrics and the baseline camera control approaches:

- *Max-min fairness without prediction of targets' motion and observation time* ($MMWP$): This algorithm is based on optimizing the reward function (5.4) without accounting for the uncertainty in the targets' motion and observation time;

[1] http://www.comp.nus.edu.sg/~lowkh/camera.html

- *M-M approach* ($M - M$): This approach is based on optimizing the expected difference between the maximum and minimum observation times of the targets. That is, the cameras are controlled such that the expected difference between the maximum and minimum observation times of the targets are minimized;
- *Round robin approach* (RRB): In this approach, all the targets are given priorities in a round robin fashion. The active cameras are controlled to observe the targets based on their priorities;
- *Maximizing coverage of targets* (COV) : In this approach, the active cameras are controlled to maximize the expected number of targets observed in their fov's (Chapter 3);
- *Systematic approach* ($AUTO$): This is a baseline approach in which the active cameras are panned to each of their states in a round robin fashion;
- *Static approach* ($STAT$): This is a baseline approach in which the active cameras are fixed at a particular state such that their fov's can maximize the coverage.

The above approaches are evaluated based on the following two performance metrics:

$$AvgMinIndex \triangleq \frac{1}{\tau}\sum_{j=1}^{\tau} \min_{k \in \mathcal{M}} t_{o_k}^{j}$$

$$AvgCoverage \triangleq \frac{1}{\tau}\sum_{j=1}^{\tau} N_{obs}^{j}$$

where τ is the total number of time steps taken in our experiments, which is set to 50, $t_{o_k}^{j}$ is the observation time of target k at time step j, and N_{obs}^{j} is the total number of targets observed by the active cameras at time step j. That is, $AvgMinIndex$ measures the average of minimum observation time over all $|\mathcal{M}|$ targets in the environment during the τ time steps and $AvgCoverage$ measures the average number of targets observed by the active cameras during the τ time steps. We will first discuss the experimental setup for the simulated experiments and analyze the results.

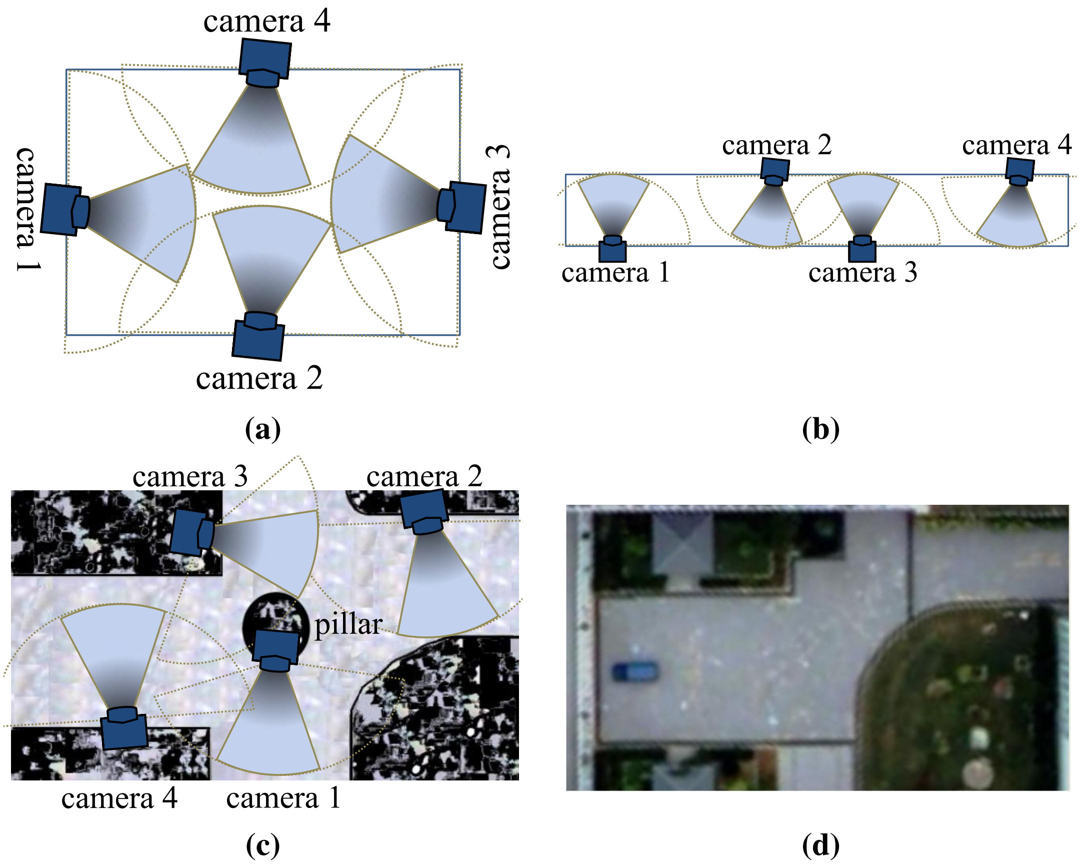


Figure 5.2: Experiment setups: (a) Hall ($|\mathcal{T}_l| = 20 \times 8$ target locations), (b) Corridor ($|\mathcal{T}_l| = 40 \times 5$ target locations), (c) Parking lot setup ($|\mathcal{T}_l| = 168$ target locations) and its corresponding real-world map in (d).

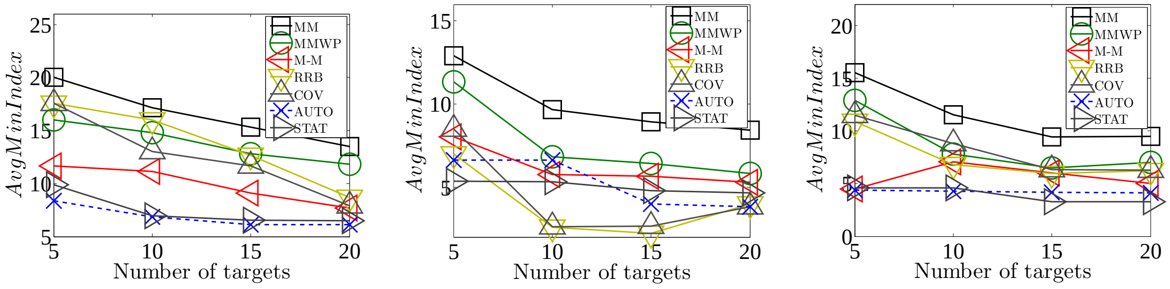
5.6.1 Simulated Experiments: Setup

The proposed approach has been extensively evaluated in the three realistic setups: (a) corridor, (b) hall, and (c) parking lot, as shown in Figure 5.2. The real-world map of the parking lot setup is shown in Figure 5.2(d). We have used the Player/Stage simulator to simulate the above surveillance environments with static cameras, active cameras, and moving targets. The simulator itself acts as the static camera that is used to observe the locations of the moving targets. The active cameras are simulated using our custom PTZ camera models that are configured across various pan angles. There are up to $m = 20$ targets whose motions are generated manually and automatically based on direction-speed motion model in the simulator as discussed in Section 5.4.2. The target's direction is taken from the following discrete set of angles {0°, 45°, . . . , -45°}. Each active camera is calibrated across its pan angles and the corresponding set $fov(c_i)$ of target locations observed by each active cameras in their states is pre-computed. In all three experimental setups, we use $n = 4$ active cameras and each active camera has $|\mathcal{C}| = 8$ states configured across pan angles. The positions and states of the cameras are chosen such that there are some target locations (known as occluded regions) in the environment, which cannot be observed by any active camera. Fairness is extremely essential in such surveillance setup because, when the target enters into those locations, the observation time of these targets will be necessarily less and hence there is a compromise in fairness in observation of targets.

5.6.2 Simulated Experiments: Results

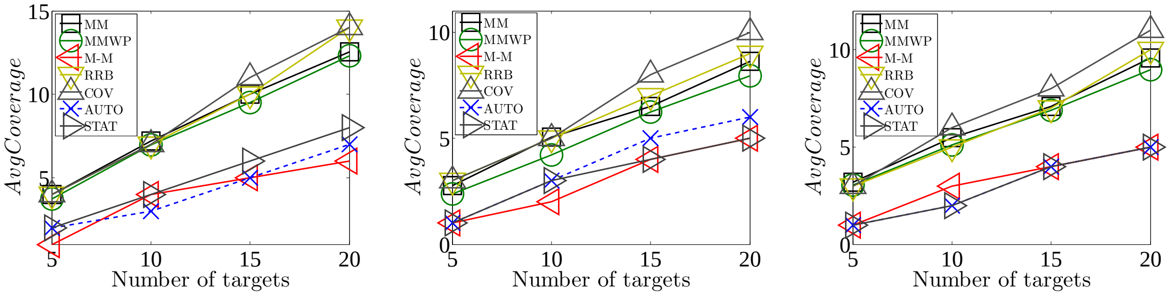
Figures. 5.3 and 5.4 show the results of the $AvgMinIndex$ and $AvgCoverage$ metrics evaluated over the three setups with $n = 4$ active cameras (each camera has $|\mathcal{C}| = 8$ states) and up to $m = 20$ targets. Our observations from the simulation results are as follows:

Uncertainty in motion and observation time: Figure 5.3 shows the $AvgMinIndex$ performance of our MDP framework based on max-min fairness metric (denoted by MM)



(a) (b) (c)

Figure 5.3: Graphs of $AvgMinIndex$ of (a) corridor, (b) hall, and (c) parking lot setup for $n = 4$, $|\mathcal{C}| = 8$, and varying number of targets $m = 5, 10, 15, 20$.



(a) (b) (c)

Figure 5.4: Graphs of $AvgCoverage$ of (a) corridor, (b) hall, and (c) parking lot setup for $n = 4$, $|\mathcal{C}| = 8$, and varying number of targets $m = 5, 10, 15, 20$.

over the $MMWP$ approach, the latter of which controls the cameras without accounting for the uncertainty of targets' motion and observation times. As seen in the result, our MM approach performs better than the $MMWP$ approach in all the three setups. This is because, in our approach, the cameras are controlled based on the predicted positions and observation times of the targets. But, in the $MMWP$ approach, the cameras are controlled based on the current location and observation times of the targets. When the targets are at the edge of the fov's of active cameras, MM approach tries to keep the targets in the center of the fov based on the predicted positions of the targets. In contrast, the $MMWP$ approach loses the targets in the next time step, which causes the loss of observation of the targets with minimum observation time. Hence, the $AvgCoverage$ of the $MMWP$ approach is also less when compared to our MM approach (see Figure 5.4). The $MMWP$ approach performs worse in the hall setup because the uncertainty of targets' motion is larger in hall setup than other setups.

Fairness in observation of the targets: Our MM approach achieves better fairness when compared to other approaches like minimizing the difference between maximum and minimum observation times of the targets ($M - M$) and round robin method (RRB) in all the three setups. In the $M - M$ approach, the cameras are controlled to minimize the difference between maximum and minimum observation times of the targets. In certain cases, this will cause the cameras not to observe any targets in order to minimize the difference. Hence, the targets with minimum observation times will be starved of observation by the cameras. Therefore, the $M - M$ approach performs poorly both in $AvgMinIndex$ (Figure 5.3) and $AvgCoverage$ (Figure 5.4). The RRB approach performs poorly in fairness as the number of targets increases (see Figure 5.3). This is because when priorities of certain targets are increased in round robin fashion, the rest of the targets are starved until their turns are reached. The COV approach, which maximizes the expected number of targets, performs poorly in the $AvgMinIndex$ metric because in order to maximize the expected number of observed targets, some targets are kept unobserved for long duration. In contrast to the above approaches, our MM approach maximizes the observation of the targets in increasing order of their observation time, i.e., the priority of a target increases with decreasing observation time. Hence, for our MM approach, the targets are not starved of observation. The baseline $AUTO$ and $STAT$ approaches suffer a lot more degradation in fairness because they do not exploit the targets' information.

Scalability in the number of targets: Figures 5.3 and 5.4 show the results for varying number of targets from $m = 5$ to 20. They reveal that our MDP framework performs better in fairness than the other approaches with an increasing number of targets and its performance degrades gracefully.

To summarize our observations from the simulation experiments,

- When our MM approach accounts for the uncertainty of targets' motion and observation time, the fairness in the observation of the targets with the least observation time will be significantly improved;
- The $M - M$ approach performs poorly in achieving the fairness because the cameras controlled by the $M-M$ approach at times try not to observe targets in order to minimize the difference between maximum and minimum observation times of the targets;
- The COV approach has a serious limitation of starving some targets of observation by active cameras for long duration;
- The RRB approach performs poorly because the low-priority targets are not observed until their turns to be observed by the active cameras are reached;
- Our MDP framework can scale for up to 20 targets in real-time and its fairness degrades gracefully as the number of targets increases.

5.6.3 Real Camera Experiments

We have tested the feasibility of deploying our MDP framework on real Axis 214 PTZ cameras. The experimental setup of our indoor lab environment is shown in Figure 5.5. The surveillance environment is of size $|\mathcal{T}_l| = 14 \times 13$ grid cells with few cells that cannot be observed by any PTZ camera (i.e., red shaded cells in Figure 5.5). We have purposely included this occluded region in our setup in order to vary the observation times of the targets when they move into these regions. There are $n = 3$ PTZ cameras, each of which has $|\mathcal{C}| = 3$ states. The fov's of the PTZ cameras are configured manually based on the resolution of the observed targets. The static camera, which is placed opposite to PTZ camera 3, can observe the entire surveillance environment at a low resolution. The static camera and 3 PTZ cameras are calibrated on a common ground plan. The targets are the

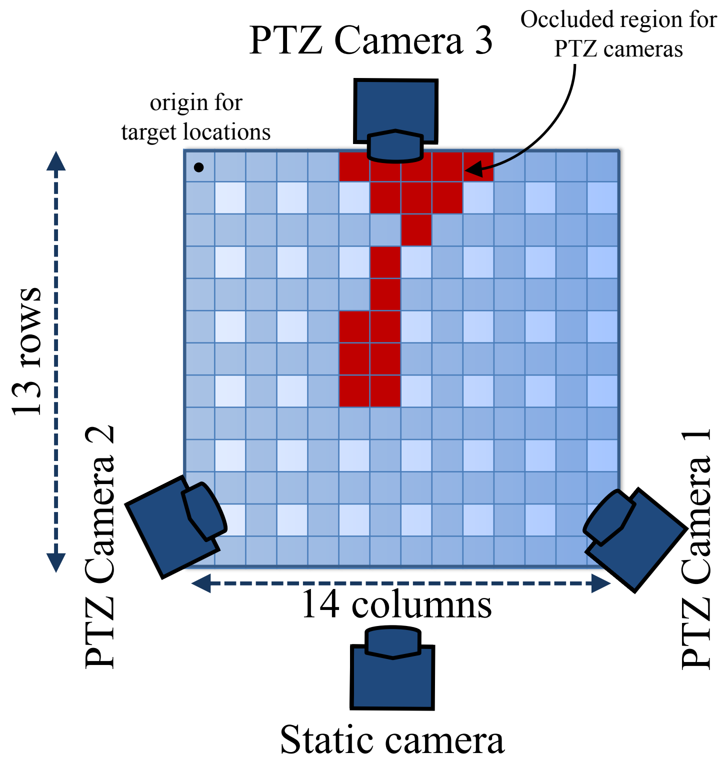
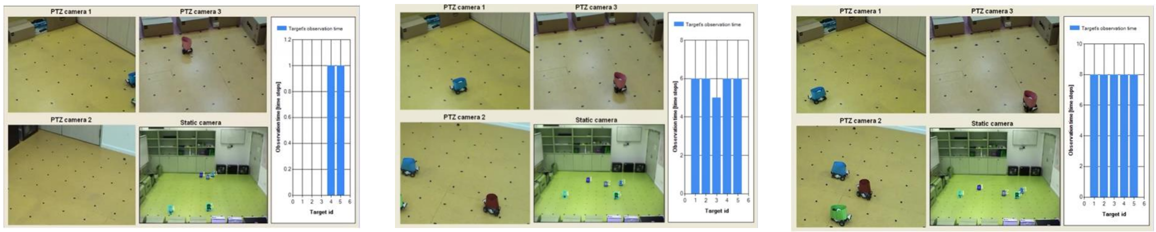


Figure 5.5: Real camera experimental setup.



Time step = 2

Time step = 7

Time step = 9

Figure 5.6: Snapshots of the observation time of $m = 5$ targets in real camera experiments.

Lego robots that are moved based on a direction-speed motion model and are programmed to turn back or stop when they hit the wall or cross other robots. The robots are tracked in the static camera based on color properties using OpenCV libraries. Using the location, direction, and speed information of the targets observed in the static camera, the 3 PTZ cameras are coordinated to observe the targets to improve fairness.

We have tested our algorithm for up to $m = 5$ targets with different interesting scenarios. For example, in the case of $m = 2$ targets, we have programmed one of the targets to move into the occluded region and wait there for a few time steps. When that target re-enters the observable region, the PTZ cameras try to focus and observe it because the observation time of that target is less than the other target. In the case of $m = 5$ targets, we have made three targets move from the occluded region to the observable region so that their observation time remain zero for a few time steps. The snapshots of the observation time of these 5 targets are shown in Figure 5.6. As mentioned, at time step 2, targets 1, 2, and 3 are inside the occluded region and hence their observation time are zero. When these targets move forward into the observable region, the cameras try to observe them and hence their observation time increase at time step 7. At time step 9, the observation time of all targets become equal and our MDP approach alternates the active cameras' observation over all the targets in order to maintain a fair observation time of them. The $AvgMinIndex$ for our real camera experiment taken over 50 time steps is shown in Table 5.2. The demo video[2] explains the interesting observations from our experiments.

Table 5.2: Performance of MDP framework in real camera experiments.

Number m of targets	2	3	4	5
$AvgMinIndex$	23	25	27	30

[2] http://www.comp.nus.edu.sg/~lowkh/camera.html

5.7 Summary

In this chapter, we have proposed a novel decision-theoretic approach to control and coordinate multiple active cameras to achieve **fairness in observation of multiple targets (FOMT)** under *fully observable* surveillance environment. As a result, the issue of starvation that plagues the existing multi-camera surveillance can be resolved. Our proposed approach is based on MDP framework that accounts for the uncertainty in the motion and observation times of the targets by modeling them probabilistically. Through our work in this chapter, the notion of fairness in multi-target observation is formally realized in the domain of multi-camera surveillance for the first time. We have exploited the conditional independence property in the targets' transition models to significantly reduce the exponential policy computation time to that of linear time in the number of targets. Empirical evaluation through simulations reveals that our approach outperforms the state-of-the-art and baseline camera coordination approaches. We have also implemented our proposed framework on real Axis 214 PTZ cameras to demonstrate its practical feasibility in a real surveillance system.

Chapter 6

MOMT Problem in Partially Observable Surveillance Environment

6.1 Introduction

In this chapter, we present a novel principled decision-theoretic approach to control and coordinate active cameras for **maximizing observations of multiple targets (MOMT)** under *partially observable* surveillance environment. That is, we do not assume that the targets' information can be observed at every time step using low-resolution static cameras as in Chapters 3, 4 and 5. In *partially observable* environment, we have only the active cameras and the targets' information are observed only through the active cameras.

Our proposed framework is the result of framing the surveillance problem formally using a rich class of decision making under uncertainty models called the *Partially Observable Markov Decision Process* (POMDP) (Section 6.2). Specifically, it resolves the issues mentioned in Chapter 1 by (a) modeling a belief of the targets' states (i.e., locations, directions, and speeds) and performing Bayesian updates of the belief (Section 6.3.4) using the probabilistic motion model of the targets (Section 6.3.2) and the probabilistic observation model of the active cameras (Section 6.3.3); (b) coordinating the active cameras' actions to simultaneously improve the belief of the targets' states and maximize the number of observed targets (Sections 6.3.5 and 6.4) while observing them at a guaranteed pre-defined resolution (Section 6.3.1); and (c) exploiting the inherent structure of our surveillance problem to improve its scalability such that it incurs linear time in the number of targets to be observed during surveillance (Section 6.4). Our proposed POMDP-based framework is empirically evaluated in simulation in various realistic surveillance environments and can achieve high-quality surveillance of up to 20 targets in real-time (Section 6.5.2). The real-world experiments with Axis 214 PTZ cameras demonstrate the practical feasibility of our POMDP-based framework in active multi-camera control and coordination (Section 6.5.3). Our POMDP framework in this chapter is different from the existing approaches and our previous work in Chapter 3 in the following ways:

- We do not assume that each and every target in the environment is completely observed at every instance in any of the cameras. We indeed model belief over the target state in order to keep track of the target when they are not observed in any of the active cameras;
- We observe targets in high-resolution active cameras in order to keep the location errors minimal;
- Most of the existing camera control approaches have serious drawbacks in scalability of number of targets. We extend our previous work in Chapter 3 to achieve scalability to increasing number of targets in a *partially observable* environments.

6.2 System Overview

The system architecture consists of POMDP controller and a supervised surveillance environment which is *partially observable*, i.e., targets cannot be always observed in the

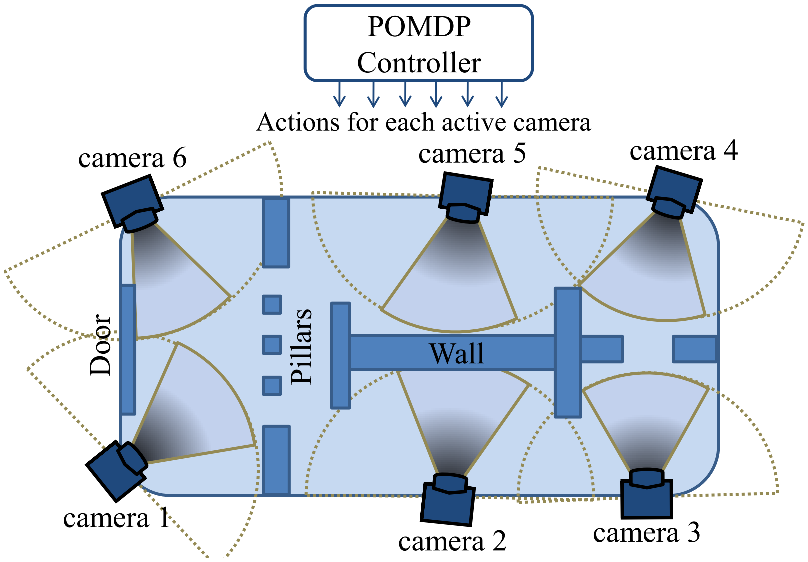


Figure 6.1: System overview of active camera network in POMDP framework.

environment due to occlusions. The environment consists of multiple targets and active cameras that are calibrated and can obtain the 3D location of targets when they are observed in their field of view. Figure 6.1 shows the top view of the surveillance environment where the full fov's of the active cameras are shown in dotted lines and the current active fov's are shaded. At any given time, the active cameras can observe only a portion of the surveillance environment. This is true in most of the real-world environments where the active cameras cannot be installed to observe the entire surveillance environment due to occlusions caused by barriers like walls, pillars, etc. This realistic setup makes the problem more challenging and practical, thus emphasizing the need for camera control framework for *partially observable* environments.

As shown in Figure 6.2, the POMDP controller models the interaction between the active cameras and the *partially observable* environment. It provides a platform to choose optimal actions for these active cameras in order to achieve high-quality surveillance. The active cameras can determine the location of the targets that are observed in its fov and passes the information to the controller. The controller determines the optimal actions of

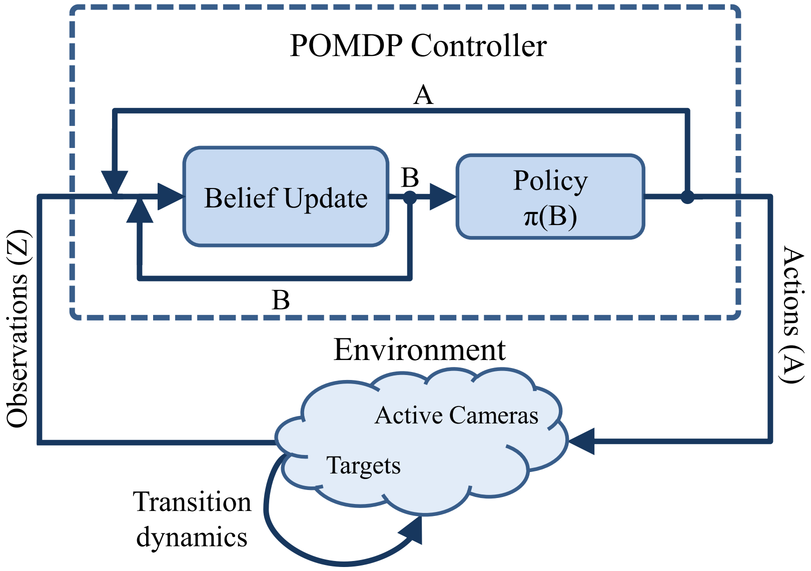


Figure 6.2: POMDP framework for controlling active cameras.

these cameras, such that the expected utility of the surveillance is maximized. The utility of the surveillance system corresponds to high-level surveillance goals that can be defined formally using real-valued objective functions as described in Section 6.3.5. The following assumptions are made in our surveillance task:

- The targets are non-evasive (i.e., they do not try to escape from the cameras' fields of view) and hence their motion cannot be controlled or influenced by the cameras.
- The targets correspondences across multiple cameras is resolved by distinct features like color, texture, etc.

Formally, a POMDP controller is defined as a tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{Z}$, T_f, O_f, R) consisting of

- a set $\mathcal{S}$ of joint states of active cameras and targets in the surveillance environment (Section 6.3.1);
- a set $\mathcal{A}$ of joint actions of active cameras (Section 6.3.1);
- a set $\mathcal{Z}$ of joint observations of the targets taken by the cameras (Section 6.3.1);
- a transition function $T_f : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denoting the probability $P(S'|S, A)$ of

going from the current joint state $S \in \mathcal{S}$ to the next joint state $S' \in \mathcal{S}$ using the joint action $A \in \mathcal{A}$ (Section 6.3.2);

- an observation function $O_f : \mathcal{S} \rightarrow [0, 1]$ denoting the probability $P(Z|S)$ of observing the joint observation $Z \in \mathcal{Z}$ given the joint state $S \in \mathcal{S}$ (Section 6.3.3); and
- a real-valued objective/reward function $R : S \rightarrow \mathbb{R}$ representing a high-level surveillance goal (Section 6.3.5).

At any given time, the exact state of the environment is not fully observable to the POMDP controller. Instead, it maintains a belief B over the set $\mathcal{S}$ of all possible states (Section 6.3.4), that is, $B(S)$ is the probability that the environment is in the state $S \in \mathcal{S}$ such that $\sum_{S \in \mathcal{S}} B(S) = 1$. As shown in Figure 6.2, at every time step, the POMDP controller issues an action $A \in \mathcal{A}$ and makes an observation $Z \in \mathcal{Z}$ from the environment. Based on the action A and observation Z, the prior belief B is updated by Bayes' rule to the posterior belief B' as follows:

$$B'(S') = \eta \, P(Z|S') \sum_{S \in \mathcal{S}} P(S'|S, A) B(S) \tag{6.1}$$

where $\eta \triangleq 1/P(Z|B, A)$ is a normalizing constant. A policy π for the POMDP controller is defined as a mapping from each belief B to an action A (Section 6.4). Solving a POMDP involves choosing the optimal policy π^* that maximizes the expected reward for any given belief B:

$$\pi^*(B) = \arg\max_{A \in \mathcal{A}} \sum_{Z' \in \mathcal{Z}} R(B') P(Z'|B, A) \, .$$

When the number of targets and active cameras increases, the state space and hence the belief space of the POMDP grow exponentially (Section 6.3.1). Therefore, computing the optimal policy incurs exponential time. Fortunately, by exploiting the structure of our surveillance problem (Sections 6.3.2 and 6.3.3), the optimal policy for a given belief B can be computed efficiently (Section 6.4).

6.3 Problem Formulation

In this section, we extend the problem formulation in Chapter 3 to POMDP framework. Similar to Section 3.3, we enumerate each components of POMDP framework and formally describe how the surveillance problem is modeled using the conventional POMDP framework. Essentially, we consider the set of targets with stochastic motion as part of the environment, and the active cameras' controller as our agent. Since these active cameras, due to their limited fields of view, may not be able to observe the whole surveillance area, the environment is only partially observable to our controller. Table 6.1 shows the summary of mathematical notations and its descriptions used in this section.

Table 6.1: Mathematical notations and its descriptions used in Chapter 6.

Notation	Description
n	Number of active cameras.
m	Number of targets to be monitored by n active cameras, such that $n \ll m$.
$\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$	State space of a target represented by a set of tuples of location, direction and speed.
$\mathcal{T}_l$	Set of all possible discretized locations of the target in the surveillance environment.
$\mathcal{T}_d$	Set of all possible discretized directions of target.
$\mathcal{T}_v$	Set of discretized speeds of the target.
t_k=$(t_{l_k}, t_{d_k}, t_{v_k})$	State of a target k is a tuple consisting of its location $t_{l_k} \in \mathcal{T}_l$, direction $t_{d_k} \in \mathcal{T}_d$ and speed $t_{v_k} \in \mathcal{T}_v$.
$T \in \mathcal{T}^m$	Joint state of m targets represented by $T = (t_1, t_2, \ldots, t_m)$.
$\mathcal{C}$	State space of a camera that consists of finite set of discretized poses of camera. Each pose of camera is given by its pan, tilt and zoom value.

$c_i \in \mathcal{C}$	State of camera i, which is given by the discretized pan, tilt and zoom value.
$C \in \mathcal{C}^n$	Joint state of n cameras represented by $C = (c_1, c_2, \ldots, c_n)$.
$fov(c_i) \subset \mathcal{T}_l$	Subset of target locations lying within the field of view (fov) of camera i in its state c_i.
$fov(C) \subset \mathcal{T}_l$	Subset of target locations lying within the joint fov of all cameras in state C, i.e. $fov(C) = \bigcup_{i=1}^{n} fov(c_i)$.
$\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$	State space of POMDP which consists set of tuples of joint state of m targets and n active cameras.
$S = (T, C) \in \mathcal{S}$	A state of POMDP that consists of joint state of m targets and n active cameras.
a_i	Action of camera i is a PTZ command to move the camera to the specified state.
A	Joint action of all active cameras represented by a tuple $A = (a_1, a_2, \ldots, a_n)$.
$\mathcal{A}$	Set of joint actions of all active cameras.
$\dot{\mathcal{Z}} = \mathcal{T}_l \cup \{\phi\}$	Set of observations of single target from active cameras which consists of possible locations of target in the environment $\mathcal{T}_l$, and a null observation ϕ when the target is not observed in any of the cameras.
$z_k \in \dot{\mathcal{Z}}$	Observation of target k from active cameras.
$\mathcal{Z} = \dot{\mathcal{Z}}^m$	Set of observations of POMDP which consists of set of tuples of joint observation of m targets.
$Z \in \dot{\mathcal{Z}}^m$	Joint observation of m targets, represented by a tuple $Z = (z_1, z_2, \ldots, z_m)$.

6.3.1 States, Actions, and Observations

A joint state $S \in \mathcal{S}$ of the POMDP controller is defined as a pair of joint states $T \in \mathcal{T}^m$ of m targets and $C \in \mathcal{C}^n$ of n active cameras where $\mathcal{T}$ and $\mathcal{C}$ denote sets of all possible states of each target and active camera, respectively. That is, $S \triangleq (T, C)$ and $\mathcal{S} = \mathcal{T}^m \times \mathcal{C}^n$. Let $T \triangleq (t_1, t_2, \ldots, t_m) \in \mathcal{T}^m$ and $C \triangleq (c_1, c_2, \ldots, c_n) \in \mathcal{C}^n$ where $t_k \in \mathcal{T}$ and $c_i \in \mathcal{C}$ denote the corresponding states of target k and camera i. Let $t_k \triangleq (t_{l_k}, t_{d_k}, t_{v_k}) \in \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$ where t_{l_k}, t_{d_k}, and t_{v_k} denote target k's location, direction, and speed, respectively. That is, $\mathcal{T} = \mathcal{T}_l \times \mathcal{T}_d \times \mathcal{T}_v$.

The state space $\mathcal{C}$ of an active camera is a finite set of discrete pan/tilt/zoom positions. Let $fov(c_i) \subset \mathcal{T}_l$ be a subset of target locations lying within the fov of camera i in its state c_i. The joint fov of all cameras in joint state C is defined as $fov(C) = \bigcup_{i=1}^n fov(c_i)$. The depth of fov of each active camera is limited such that (a) imageries of the targets detected within its fov satisfy a pre-defined resolution, and (b) the observed locations of the targets detected within its fov are of minimal location error. This is done by adjusting the zoom parameter of each camera based on its position.

The joint actions of the POMDP controller are PTZ commands that move the corresponding cameras to their specified states. Let a joint action of the n cameras be denoted by $A \triangleq (a_1, a_2, \ldots, a_n) \in \mathcal{A}$ where a_i denotes the PTZ command of camera i.

Let $\dot{\mathcal{Z}} \triangleq \mathcal{T}_l \cup \{\phi\}$ denote a set of all possible observations of a target comprising the set $\mathcal{T}_l$ of all possible locations of the target in the environment and a null observation ϕ when the target is not observed by any of the cameras. Let an observation of target k be denoted by $z_k \in \dot{\mathcal{Z}}$ and a joint observation of the m targets be denoted by $Z \triangleq (z_1, z_2, \ldots, z_m) \in \dot{\mathcal{Z}}^m$. That is, $\mathcal{Z} = \dot{\mathcal{Z}}^m$.

6.3.2 Transition Model T_f

By exploiting the following structural assumptions in the state transition dynamics of the surveillance environment:

- camera i's next state c'_i is conditionally independent of the other $n-1$ cameras' states and actions and the m targets' states given its current state c_i and action a_i for $i = 1, \ldots, n$

and

- target k's next state t'_k is conditionally independent of the n cameras' states and actions (i.e., target's motion is not affected by the cameras' states and actions) and the other $m-1$ targets' states (i.e., every target moves independently) given its current state t_k for $k = 1, \ldots, m$,

the transition model T_f can be factored into transition models of individual targets and active cameras, hence significantly reducing the time incurred to compute the optimal policy π^* for a given belief B (Section 6.4). Furthermore, since the modern active cameras are able to move to their specified positions accurately [Axi, 2011], it is practical to assume the transition model of each individual camera to be deterministic and consequently represented by a function τ that moves camera i from its current state c_i to its next state $\tau(c_i, a_i)$ by the action a_i. Then, the transition model of the POMDP controller can be simplified to

$$P(S'|S,A) = \prod_{k=1}^{m} P(t'_k|t_k) \prod_{i=1}^{n} \delta_{\tau(c_i,a_i)}(c'_i) \tag{6.2}$$

where $\delta_x(x')$ is a Kronecker delta function of value 1 if $x' = x$, and 0 otherwise. Details on the derivation of (6.2) are reported in Section 3.3.2 in Chapter 3. The state transition of target k from t_k to t'_k includes stochastic transitions of its location from t_{l_k} to t'_{l_k}, its direction from t_{d_k} to t'_{d_k}, and its speed from t_{v_k} to t'_{v_k}. So, the transition probability of target k can be factored into transition probabilities of its location, direction, and speed:

$$P(t'_k|t_k) = P(t'_{l_k}|t_{l_k}, t'_{d_k}, t'_{v_k})P(t'_{d_k}|t_{d_k})P(t'_{v_k}|t_{v_k})\ .$$

The transition probabilities $P(t'_{d_k}|t_{d_k})$ and $P(t'_{v_k}|t_{v_k})$ of the target's direction and speed are, respectively, modeled as Gaussian distributions $\mathcal{N}(\mu_d, \sigma_d)$ and $\mathcal{N}(\mu_v, \sigma_v)$ with the means μ_d and μ_v being the current direction and speed of the target, and σ_d and σ_v being the variance parameters which are learned from a dataset of the targets' trajectories in the environment. The transition probability $P(t'_{l_k}|t_{l_k}, t'_{d_k}, t'_{v_k})$ of the target's next location is constructed using the general direction-speed motion model, as described in Section 3.3.2.1 in Chapter 3.

6.3.3 Observation Model O_f

Similar to the factorization of the transition model T_f, the observation model O_f can also be factored into observation models of individual targets using the following structural assumption: The observed location $z_k \in \dot{\mathcal{Z}}$ of target k is conditionally independent of the observed and true states of the other $m-1$ targets and its true direction $t_{d_k} \in \mathcal{T}_d$ and speed $t_{v_k} \in \mathcal{T}_v$ given its true location $t_{l_k} \in \mathcal{T}_l$ and the joint state $C \in \mathcal{C}^n$ of the n active cameras for $k = 1, \ldots, m$. As a result, the time incurred to compute the optimal policy π^* for a given belief B can be significantly reduced (Section 6.4). Then, the observation model of the POMDP controller can be simplified to

$$P(Z|S) = \prod_{k=1}^{m} P(z_k|t_{l_k}, C) \ . \tag{6.3}$$

The derivation of (6.3) is reported in Appendix A.4. The observation probability $P(z_k|t_{l_k}, C)$ of target k depends on whether the target lies within the joint fov of the active cameras. When the target lies within the cameras' joint fov corresponding to their joint state C (i.e., $z_k \neq \phi$), the observation model of target k becomes deterministic:

$$P(z_k|t_{l_k}, C) = \begin{cases} 1 & \text{if } z_k = t_{l_k} \wedge t_{l_k} \in fov(C), \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, when target k does not lie within the joint fov of the active cameras corresponding to their joint state C, the observation probability of target k is uniformly distributed over the locations not covered by the joint fov (i.e., $\overline{fov(C)}$):

$$P(z_k = \phi | t_{l_k}, C) = \begin{cases} \dfrac{1}{|\overline{fov(C)}|} & \text{if } t_{l_k} \notin fov(C), \\ 0 & \text{otherwise.} \end{cases}$$

6.3.4 Bayesian Belief Update

By making use of independence assumptions similar to that in the transition model (Section 6.3.2), a belief B can be factored into beliefs of individual targets and cameras:

$$\begin{aligned} B(S) &= P((T, C)) = P(T)P(C) \\ &= \prod_{k=1}^{m} P(t_k) \prod_{i=1}^{n} P(c_i) = \prod_{k=1}^{m} b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}_i}(c_i) \end{aligned} \tag{6.4}$$

where b_k denotes a belief over the set $\mathcal{T}$ of all possible states of target k (i.e., $b_k(t_k)$ is the probability that target k is in state t_k) and $\hat{c}_i$ is the current state of camera i that, unlike a target's state, is fully observable to the POMDP controller since its position can be directly read from its port. Hence, the probability $P(c_i)$ of a state c_i of camera i can be represented by a Kronecker delta $\delta_{\hat{c}_i}(c_i)$ and the last equality in (6.4) follows.

The POMDP controller issues a joint action A to move each camera i from current state $\hat{c}_i$ to next state $\hat{c}'_i$, receives an observation z_k of each target k, and then updates the prior belief B to the posterior belief B' using Bayes' rule (6.1). Similar to the factorization of the prior belief B above, the posterior belief B' can also be factored into posterior beliefs of individual targets and cameras:

$$B'(S') = \prod_{k=1}^{m} b'_k(t'_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \tag{6.5}$$

where the posterior belief b'_k of target k is defined as

$$b'_k(t'_k) \triangleq \eta_k P(z_k|t'_{l_k}, C') \sum_{t_k \in \mathcal{T}} P(t'_k|t_k) b_k(t_k) \ , \tag{6.6}$$

$C' \triangleq (c'_1, \ldots, c'_n)$, and $\eta_k \triangleq 1/P(z_k|b_k, C')$ is a normalizing constant. The derivation of (6.5) is reported in Appendix A.5.

6.3.5 Objective/Reward Function R

The goal of the surveillance system is to maximize the number of targets observed with a guaranteed resolution. This can be achieved by defining a reward function that measures the total number of targets lying within the joint fov of the active cameras corresponding to their joint state C:

$$R(S) = R((T, C)) \triangleq \sum_{k=1}^{m} \widetilde{R}(t_k, C) \tag{6.7}$$

where

$$\widetilde{R}(t_k, C) \triangleq \begin{cases} 1 & \text{if } t_{l_k} \in fov(C), \\ 0 & \text{otherwise.} \end{cases}$$

Since the exact locations of the targets may not be fully observable to the cameras at all times, the POMDP controller has to track the joint belief B of the targets and consider the *expected* reward with respect to this belief instead:

$$R(B) \triangleq \sum_{S \in \mathcal{S}} R(S) B(S) = \sum_{k=1}^{m} \widetilde{R}(b_k, \widehat{C}) \tag{6.8}$$

where $\widehat{C} \triangleq (\hat{c}_1, \ldots, \hat{c}_n)$ and

$$\widetilde{R}(b_k, C) \triangleq \sum_{t_k \in \mathcal{T}} \widetilde{R}(t_k, C) b_k(t_k) \ . \tag{6.9}$$

The derivation of (6.8) is reported in Appendix A.6.

6.4 Policy Computation

Recall that a policy π for the POMDP controller is a mapping from each belief B to a joint action $A \in \mathcal{A}$ of the n cameras. At every time step, the POMDP controller determines the optimal policy π^* for the belief B such that the expected number of observed targets in the next time step is maximized. Since the observations of the m targets taken by the cameras in the next time step are not known to the POMDP controller, it has to consider the *expected* reward with respect to these future observations. Then, the optimal policy π^* for a given belief B becomes

$$\pi^*(B) = \arg\max_{A \in \mathcal{A}} V(B, A) \tag{6.10}$$

where

$$V(B, A) = \sum_{Z \in \mathcal{Z}} R(B') P(Z|B, A) \ . \tag{6.11}$$

Computing the policy π^* (6.10) for a given belief B incurs $\mathcal{O}(|\mathcal{A}||\dot{\mathcal{Z}}|^m|\mathcal{T}|)$ time which is exponential in the number m of targets. Fortunately, by exploiting simplified transition and observation models due to conditional independence assumptions (i.e., (6.2) and (6.3)), this computational cost can be significantly reduced. In particular, it is derived in Appendix A.7 that the value function $V(B, A)$ of m targets can be simplified to comprise a sum of value function $\widetilde{V}(b_k, C')$ of individual target k for $k = 1, \ldots, m$:

$$V(B, A) = \sum_{k=1}^{m} \widetilde{V}(b_k, (\tau(\hat{c}_1, a_1), \ldots, \tau(\hat{c}_n, a_n))) \tag{6.12}$$

where

$$\widetilde{V}(b_k, C') \triangleq \sum_{z_k \in fov(C')} \sum_{t'_k \in \mathcal{T}} \widetilde{R}(t'_k, C') \hat{b}'_k(t'_k) \tag{6.13}$$

and $\hat{b}'_k$ is the unnormalized belief of target k (i.e., $\hat{b}'_k(t'_k) = b'_k(t'_k)/\eta_k$). Using (6.12) and (6.13), we obtain the following result:

Theorem 4. If (6.2) and (6.3) hold, then computing policy π^* (6.10) for a given belief B incurs $\mathcal{O}(|\mathcal{A}||\dot{\mathcal{Z}}||\mathcal{T}|m)$ time.

Computing the value function $\widetilde{V}(b_k, C')$ (6.13) for a single target k incurs $\mathcal{O}(|\dot{\mathcal{Z}}||\mathcal{T}|)$ time. For m targets, the value function $V(B, A)$ (6.12) therefore incurs $\mathcal{O}(|\dot{\mathcal{Z}}||\mathcal{T}|m)$ time. Finally, computing the optimal policy π^* (6.10) for a given belief B incurs $\mathcal{O}(|\mathcal{A}||\dot{\mathcal{Z}}||\mathcal{T}|m)$ time which is linear in number m of targets.

6.5 Experiments and Discussion

This section evaluates the performance of our proposed POMDP controller over the existing approaches in *partially observable* environments. We have used Player/Stage simulator [Gerkey *et al.*, 2003] to evaluate the proposed approach and implemented in real Axis 214 PTZ cameras to show the feasibility in real-camera surveillance. Its important to note that the existing camera control approaches in the literature are proposed for the *fully observable* environments where the states of the targets are always observed by either static cameras or PTZ cameras in wide-view (see Table 2.2). To make the evaluation fair, we compare these works in a *partially observable* environments. Our POMDP approach (denoted as P

in Figure 6.4, 6.5, and 6.6) which uses only active (i.e. PTZ) cameras is compared against the following existing approaches:

- *MDP with only PTZ cameras (MP):* This approach uses MDP framework in Chapter 3 to control and coordinate the active cameras. There are no static cameras to directly observe the targets' locations and hence they are observed only from the active cameras' fov.
- *MDP with Static and PTZ cameras (MSP):* This approach uses MDP framework in Chapter 3 to control and coordinate the active cameras that are supported by wide-view static cameras. These static cameras are used to observe the targets' location, direction and speed information. A Gaussian noise is added to the location of each target observed by the static cameras such that the Gaussian variance increases with greater distance of the target from the static camera.
- *Systematic Approach (Sys):* The active cameras are panned automatically to each of its states in a round robin fashion for every time step.
- *Static Approach ($Stat$):* The active cameras are fixed at a particular state such that they observe maximum area of the environment with the pre-defined image resolution.

We use the same performance metric $PercentObs$ as in Chapter 3 to evaluate the above approaches. It is given as

$$PercentObs = \frac{100}{\tau M_{tot}} \sum_{i=1}^{\tau} M_{obs}^{i}$$

where τ (i.e., set to 100 in simulations) is the total number of time steps taken in our experiments, M_{obs}^{i} is the total number of targets observed by the active cameras at a given time step i, and M_{tot} is the total number of targets present in the environment. That is, the $PercentObs$ metric averages the percentage of targets being observed by the active cameras over the entire duration of τ time steps. First we will discuss the experimental setups followed by detailed analysis of experimental results. Then we will show the results

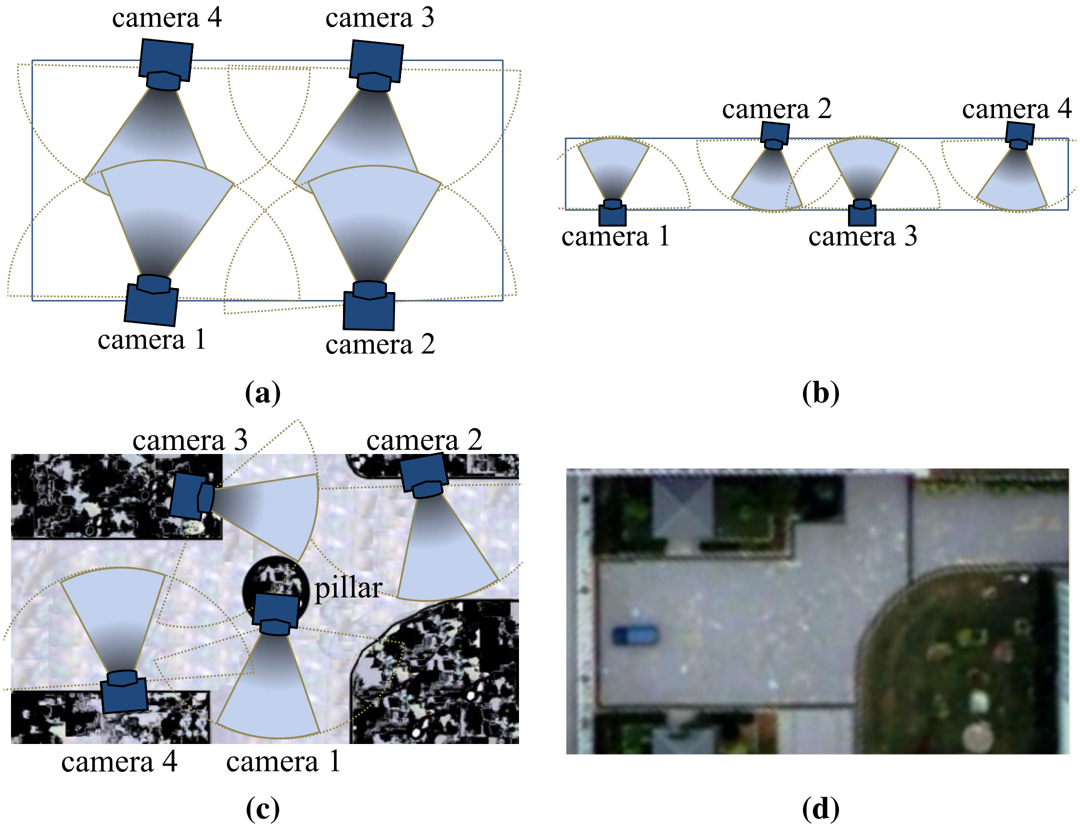


Figure 6.3: Experiments setups: (a) Hall ($|\mathcal{T}_l| = 20 \times 10$ target locations), (b) Corridor ($|\mathcal{T}_l| = 40 \times 5$ target locations), (c) Parking lot setup ($|\mathcal{T}_l| = 168$ target locations) and its corresponding real-world map in (d).

of real camera experiments. Interested readers can look into our demo video[1].

6.5.1 Simulated Experiments: Setup

The proposed approach is evaluated against the above approaches in three different setups as shown in Figure 6.3: (a) a corridor ($|\mathcal{T}_l| = 40 \times 5$ target locations), (b) a hall ($|\mathcal{T}_l| = 20 \times 10$ target locations) and (c) a parking lot setup ($|\mathcal{T}_l| = 168$ target locations). As mentioned in Chapter 5, the parking lot setup has been taken from the real world map (as

[1] http://www.comp.nus.edu.sg/~lowkh/camera.html

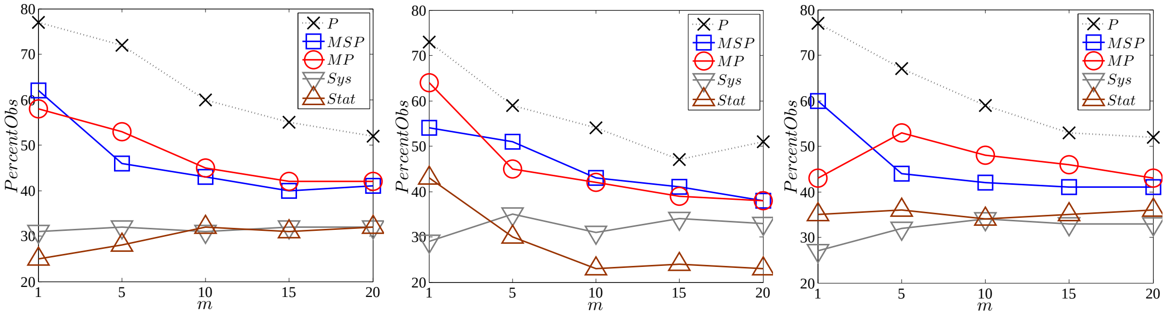
shown in Figure 6.3(d)) of our university space which consists of obstacles (black shades in Figure 6.3(c)) like buildings, walls, etc. In order to introduce more occlusions in the environment, we have added a virtual pillar in the center of the parking lot setup (Figure 6.3(c)).

The active cameras are simulated in Player/Stage simulator by configuring the states of the cameras across the pan angles as discussed in Section 6.3.1. There are $n = 4$ active cameras with $|\mathcal{C}| = 3$ states each. These position and states of the cameras are chosen such that there are some target locations in the environment which cannot be observed by any of the active cameras, in order to generate the *blind regions*. For example, in corridor setup (Figure 6.3(b)), the cameras 1 and 2 are separated far apart, so that when the targets move in between these cameras, the targets are not observed. This is true in many real world environments and simulates the partial observability in the experimental setups. The targets' trajectories are generated as discussed in Section 3.5.1 in Chapter 3.

6.5.2 Simulated Experiments: Results

Figure 6.4 and 6.5 show the comparison of performance of different approaches for up to m=20 targets for all the three setups. It can be seen from the graphs, our POMDP approach (denoted as P in the graph) outperforms other approaches in all three camera setups. The detailed observations from the experiments are as follows:

The observations from our experiments with MP approach are as follows: (a) Firstly, when the targets leave the fov of any of the cameras and enters the *blind regions*, the MP approach have no idea on where the targets will be moving in the next few time steps. This is because, the MP approach can control the cameras to focus the targets only when the states of the targets are observed in any of the active cameras, and have no notion of the targets that are not observed in any of the cameras. In contrast, our POMDP based approach predicts the belief of the targets based on the targets' transition model and updates the belief based on the cameras' observation model. So when the targets leave the fov of the



(a) **(b)** **(c)**

Figure 6.4: Graphs of $PercentObs$ vs. number m of targets with n=4 active cameras, whose $|fov(c_i)| \approx 14$ cells for the environments: (a) corridor, (b) hall and (c) parking lot.

cameras and enters the *blind region*, the belief of the targets are updated at every time step. Since in our POMDP approach, we determine actions of the cameras based on the expected belief of the targets, the cameras are panned in advance to observe the incoming targets. (b) Secondly, the MP approach determines the cameras' action based on the locations and directions of the targets that are observed in the active cameras. When the targets enters into any of the active cameras from the *blind regions*, the direction of the targets are wrongly interpreted by the MDP controller. This is one of the serious limitation of the MP approach where there is no notion of knowing the direction of the targets when they are in the *blind region*. Whereas in our POMDP approach, the direction of the targets are modeled by its transition model in the belief update step. Thus belief filtering process in our POMDP approach helps in tracing the locations and directions of the targets, even when they are not observed in any of the cameras.

The observations from our experiments with MSP approach are as follows: When the static cameras observes the targets that are far away, they obtain the noisy locations of the targets. This in turn induce the errors in direction and speed of the targets. In MSP approach, the active cameras are controlled to observe the predicted locations of the targets

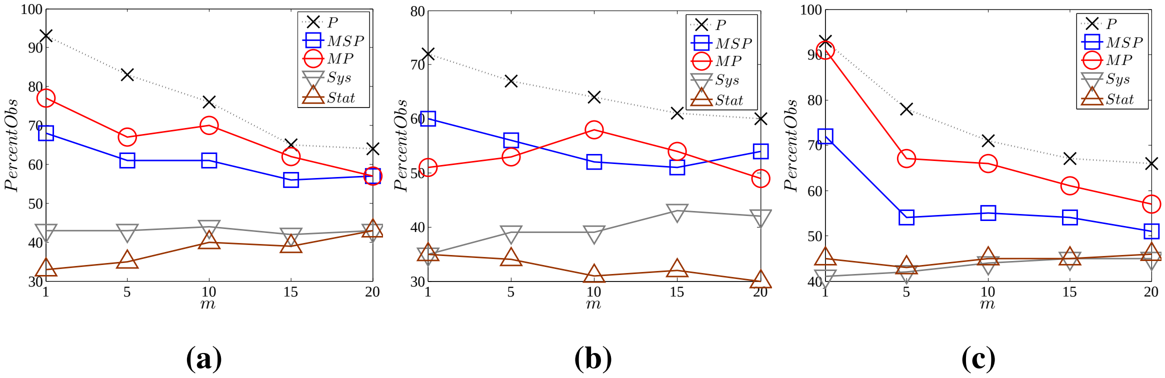


Figure 6.5: Graphs of $PercentObs$ vs. number m of targets with n=4 active cameras, whose $|fov(c_i)| \approx 20$ cells for the environments: (a) corridor, (b) hall and (c) parking lot.

in the next time step, based on the targets' information (i.e., location, direction and speed) obtained from the static cameras. When the noisy targets' information are used in MDP controller, it causes miss predictions in the expected locations of the targets, which consequently affects the performance of the MSP approach. In contrast, in our POMDP based approach, the targets locations are observed at a high-resolution active cameras whose calibration error is bounded by limiting the depth of its fov (see Section 6.3.1). Since the observations (i.e., locations of the targets) for POMDP are more accurate than in MSP approach, the predictions of the targets' locations and directions through belief filtering process are also accurate. Hence, our POMDP approach outperforms the MSP approach as shown in the Figure 6.4 and 6.5.

The observations from our experiments with Sys and $Stat$ approaches: Our POMDP approach performs much better than the Sys and $Stat$ baseline approaches because, in our approach, the active cameras are controlled based on the targets' predicted motion and the observations from the active cameras. Whereas in Sys approach, the cameras are panned without accounting the targets information like locations, direction, etc., and in $Stat$ approach, the cameras are placed static in one of the states.

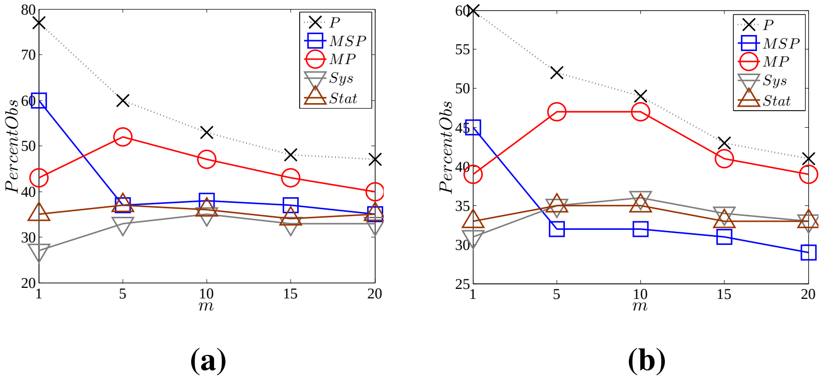


Figure 6.6: Graphs of $PercentObs$ vs. number m of targets in parking lot setup with n=4 active cameras with (a) camera 1 is made static and (b) camera 1 and 3 are made static in one of their respective states.

We have tested the robustness of our approach when one or more active cameras fails to do pan/tilt/zoom operations and may get stuck in one of its state. This can happen in real surveillance systems due to hardware malfunctioning of active cameras. We have evaluated our approach by keeping one or more active cameras to be static in one of the states. Figure 6.6(a) shows the results of having camera 1 as static in the parking lot setup and Figure 6.6(b) shows the results of having both camera 1 and camera 3 to be static in one of their states. From these results, we can see that the performance of our approach is much better than the other approaches because: (a) When one or more active cameras gets stuck on any of their states, the number of target locations that can be observed by these cameras by pan/tilt/zoom operations, will be reduced significantly. Since our POMDP approach has the ability to keep track of the unobserved targets' locations and directions through its belief filtering process, it performs better than other approaches. (b) In our POMDP approach, when the targets pass through camera 1 or 3 in the parking lot setup, the belief filtering process predicts the incoming targets and hence the camera 2 or 4 is panned in advance to observed these targets.

We have also experimented our approach with existing approaches for different resolution of the active cameras. The resolution of the active cameras in the simulator are adjusted by modifying the fov polygon. When the average fov of active cameras is decreased from

$|fov(c_i)| \approx 20$ (Figure 6.5) to $|fov(c_i)| \approx$14 (Figure 6.4), the over all performance of our approach is much better than other approaches. This is because, when the targets are observed at high-resolution (i.e., $|fov(c_i)| \approx 14$), then the number of targets' locations that are not observed by any of the active cameras increases. In such situations, our POMDP approach performs better than other approaches because, the POMDP approach is aware of the targets locations and directions through its belief filtering process even when they are not visible in any of the active cameras.

We summarize the observations from our simulation experiments:

- Our POMDP based approach performs better than MP approach because of the ability of POMDP to keep track of targets' locations and directions through its belief update process.
- The POMDP approach outperforms MSP approach because the observations (i.e., target's location) made from active cameras in high-resolution in our POMDP approach is accurate when compared to the noisy observations from static cameras of MSP approach.
- The Sys and $Stat$ approaches suffers performance degradation because, the cameras are controlled independent of targets' information.
- Our POMDP approach is robust even when one or more active cameras get struck in one of their state.
- The performance of our POMDP approach is much better than other approaches when the resolution of the active cameras is increased.

6.5.3 Real Camera Experiments

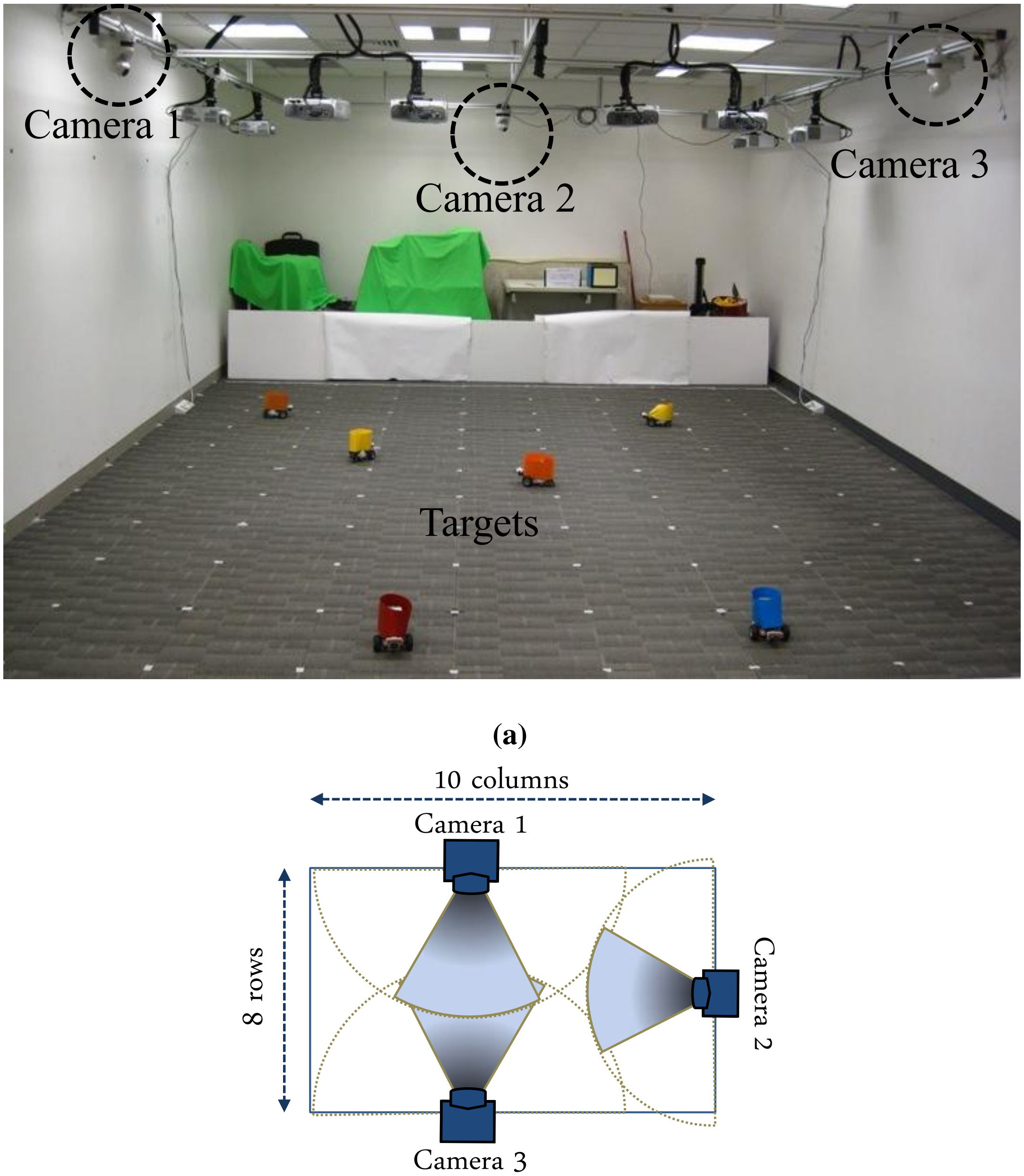
We have tested the feasibility of our POMDP controller in real Axis 214 PTZ cameras in order to monitor Lego robots (targets) over the environment of size $|\mathcal{T}_l| = 10 \times 8$ grid cells. We have $n = 3$ PTZ cameras whose number of states of each camera is $|\mathcal{C}| = 3$.

The cameras are placed such that, at any given time, only a subset of targets' locations are observed in the active cameras. The experimental setup for our real camera experiments is shown in Figure 6.7. These cameras are calibrated in each of its state [Tsai, 1986] and the depth of the fov of these cameras are determined empirically for each of the cameras. The Lego robots are programmed to move based on the direction-speed motion model. The transition model, observation model and objective function for a single target are computed and are stored off line for the above setup.

Individual videos from each of the cameras are processed using OpenCV image processing libraries in order to detect and differentiate the targets in its view. Initially, background images for each states of the active cameras are captured and the color histogram of individual targets are stored off line. We call this color histogram of the targets as knowledge base. The robots are mounted with color markers to support the target detection and recognition process. We have used color histogram of the robots to differentiate and discriminate the targets. Targets are detected using the background subtraction and morphological operations. The target recognition is done by matching the color histogram of the targets with the color histogram of the robots in the knowledge base. We use Bhattacharyya distance to match their color histograms.

The POMDP controller is initialized with the initial belief of the robots based on the locations of the targets. In every time step, we capture the images from all the active cameras and process to detect and recognize the targets and their observed 3D location. The belief of individual robots are updated based on observation of each robot from the active cameras and the cameras' action. The state of the cameras are directly read from the individual cameras. Based on the belief of all the targets, we compute the optimal actions for the cameras, such that expected number of targets in maximized. Table 6.2 shows the performance of our approach in real camera experiments. Visual results of our real-camera experiments are demonstrated in our video[2]

[2] http://www.comp.nus.edu.sg/~lowkh/camera.html



(a)

(b)

Figure 6.7: (a) Real experiment setup containing three Axis 214 PTZ cameras (marked with dotted circle), the colored Lego targets on the surveillance environment of size 10×8 grid cells. (b) Corresponding overhead view of the setup.

Table 6.2: Performance of POMDP framework in real camera experiments.

Number m of targets	1	2	3	4	5
$PercentObs$	98.2	96.6	93.3	91.5	87

6.6 Summary

In this chapter, we have presented a novel decision-theoretic approach to control and coordinate multiple active cameras in order to maximize the number of targets observed in an uncertain and *partially observable* surveillance environments. Specifically, we have designed a POMDP framework that helps to eliminate the dependency of wide-view static cameras for tracking targets' location, and simultaneously performs the tracking and observing targets at high-resolution. We have exploited the conditional independence property of targets' motion and observation in our surveillance problem, in order to reduce the exponential policy computation to linear in increasing number of targets. The experimental evaluation shows that our proposed camera control approach is robust and performs better than the existing approaches . The real experiments in Axis 214 PTZ cameras show the practicality of our approach in real surveillance systems.

Chapter 7

Conclusions and Future Works

In this thesis, we have studied the following central problem in surveillance:

> How can a network of active cameras be coordinated to monitor a set of moving targets with a guaranteed image resolution?

7.1 Summary of Contributions

To address the above question in surveillance, this thesis has provided the following novel contributions:

1. **Decision-theoretic formulation** [Natarajan, 2012a; Natarajan, 2012b]**:** We have presented a novel decision-theoretic multi-agent formulation to control and coordinate multiple active cameras in surveillance. We have modeled the surveillance task as stochastic optimization problem, where the active cameras are controlled and coordinated to achieve the desired surveillance goals in presence of uncertainties. Our proposed decision-theoretic frameworks, coordinates active cameras under *fully observable* and *partially observable* surveillance environments. Specifically, we have

provided decision-theoretic formulations for two novel problems in active camera surveillance: **maximizing observations of multiple targets (MOMT)** in *fully observable* and *partially observable* surveillance environments; and **fairness in observation of multiple targets (FOMT)** in *fully observable* surveillance environment.

2. **MDP framework for MOMT** [Natarajan *et al.*, 2012a]**:** We have proposed a novel, principled MDP framework that accounts for the targets' motion uncertainty through probabilistic motion model. The trade-off between the number of targets observed and the resolution of observing them is addressed by coordinating the actions of the cameras to maximize the expected number of targets with guaranteed resolution. In particular, the fov of active cameras are adjusted to guarantee the desired resolution of targets. By exploiting the conditional independence property of transition model in our surveillance problem, we reduced the exponential policy computation time to linear time in number of targets. Therefore, our MDP framework can achieve high-quality surveillance of up to 50 targets in real-time. We have also found that our greedy solution is equivalent to finite horizon planning solution, when the transition model of the active cameras is deterministic.

3. **MDP framework for MOMT in large-scale camera network:** We have extended the MDP framework for MOMT task to large camera networks using the concepts of coordination graph and max-plus algorithm. We have achieved computational efficiency by decomposing the global coordination problem into set of pair-wise local coordination problems, approximately, using the coordination graph. Then we have used max-plus algorithm to solve the local coordination problems efficiently, through message passing mechanism. Our experimental results show that the proposed coordination approach performs better than the other baseline approaches even for 50 cameras and 60 targets. Our experiments with 6 cameras and varying number of cycles in the coordination graph shows that our solution is closer to the centralized

MDP solution when there are less cycles in the coordination graph.

4. **MDP framework for FOMT:** We have proposed a novel, principled MDP framework that optimizes the max-min fairness metric to control and coordinate active cameras in order to observe the targets fairly. As a result, the issue of "starvation" of observation of targets by active cameras is resolved. Through this work, the notion of fairness in multi-target observation has been realized formally in the domain of multi-camera surveillance for the first time. The uncertainty in motion and observation times of the targets is model by probabilistic transition model. By exploiting the conditional independence property of the transition model, we have reduced the exponential policy computation time to linear in number of targets. Our simulation results shows that our proposed framework outperforms the state-of-the-art camera control approaches and other baseline approaches. Our real camera experiments demonstrates the practical feasibility of our approach and notion of fairness in real surveillance system.

5. **POMDP framework for MOMT** [Natarajan *et al.*, 2012b]**:** We have proposed a novel, principled POMDP framework for MOMT problem in *partially observable* surveillance environment. This framework models the belief of the targets' states and performs Bayesian updates of the belief using probabilistic transition model of targets and probabilistic observation model of cameras. Specifically, our approach helps to eliminate the strong dependency of wide-view static cameras to track the targets' locations and simultaneously performs the tracking and observation of targets at high-resolution. We have exploited the conditional independence property in the transition model and the observation model to reduce the exponential policy computation time to linear in number of targets. Our experiments show that our POMDP approach can perform better than the other approaches in *partially observable* environments. We have also demonstrated that our approach is robust and can

perform better than other approaches even when one or more active cameras fail to pan/tilt/zoom due to hardware malfunctioning. The real camera experiments show the practicality of our POMDP approach in real surveillance environment.

7.2 Limitations and Future Works

This section enumerates some of the future works that can be continued along the direction of this thesis.

1. **Parallelizing the MDP framework based on Max-plus algorithm.** In Chapter 4, our MDP framework has been implemented in a centralized manner. In order to avoid central point failure and make each camera to coordinate autonomously, we will parallelize our MDP framework using MPI or Hadoop parallel programming framework. We would like to improve the quality of the solution further, by removing cycles in the coordination graph and dynamically constructing it at every time step based on the targets' information. When there is no target that can be observed in a camera's possible fov, then the edges to that camera can be removed in the coordination graph. Alternatively, a real valued weight can be added to each edge of the coordination graph and remove the edges with low weights in order to remove cycles in the coordination graph. The weights can be calculated based on the expected observation of targets in each active cameras.

2. **Improving scalability in number of cameras for MDP framework to solve FOMT task.** Currently, our MDP framework in Chapter 5 can scale well for increasing number of target, but has its limitation in scalability in number of cameras. Therefore, we will extend our work in Chapter 4 to improve scalability in number of cameras for FOMT surveillance task.

3. **Addressing the trade-off between coverage and fairness in the active camera surveillance.** Our work in Chapter 5, addresses only the fairness in observation of targets. When these cameras are controlled to improve only fairness in observation of targets, there is a compromise in coverage of active cameras. On the other hand, when the cameras are controlled to maximize only the coverage or the number of targets as in Chapter 3, then there is lack of fairness in observation of targets. We will investigate and model this interesting and non-trivial trade-off between fairness and coverage of active cameras in future. Particularly, this can done in two steps: First, determine the joint actions of cameras that guarantees certain coverage or expected number of targets to be observed in active cameras. In the next step, use the same joint actions of cameras to improve the fairness. The resultant optimal action will guarantee certain coverage and at the same time improves fairness.

4. **Other fairness metrics for active camera surveillance.** Our fairness metric defined in Chapter 5 has advantage of being reduced to an expression that can be computed efficiently. Nevertheless, there can be other fairness metrics in the literature that can be used in surveillance. We will investigate other metrics like entropy of observation time, average throughput of surveillance, etc. to measure fairness in observation of targets.

5. **Improving scalability in number of cameras for POMDP framework to solve MOMT task.** Our work in Chapter 6, scales well only in number of targets and scales poorly with increase in number of cameras. Therefore, we will investigate on how to use the coordination graph concepts to decompose the centralized coordination problem in *partially observable* environment into set of local coordination problems and solve them approximately using max-plus algorithm.

6. **Study more realistic motion models for human motion and test in our MDP and POMDP frameworks.** In this thesis, the transition probabilities of targets are

computed based on simple direction-speed motion model that follows Gaussian distribution for target's direction and speed. But in practice, real human motions in the surveillance environment can be more complex and may need a better motion model. We would like to explore on more recent works in human motion models like human steering model [Tastan, 2013], human motion predictions from social forces [Luber *et al.*, 2010], etc. to calculate our target's transition probabilities.

7. **Model uncertainty in vision algorithms.** In the whole thesis, we have assumed that the vision algorithms for tracking and recognizing targets are near perfect which is not always true in real world environments. In future, we would like to account for the uncertainties in the underlying vision algorithms in our coordination framework. One possible way to encode these uncertainties is to modify the observation model of our POMDP framework in Chapter 6 based on the underlying vision algorithms.

Appendix A

Proofs

A.1 Value function decomposition in MDP framework for MOMT problem

The value function V (3.7) is given by

$$\begin{aligned}
V(T, C, A) &= \sum_{T' \in \mathcal{T}^m} R((T', C'))\ P(T'|T) \\
&= \sum_{t'_1 \in \mathcal{T}, \ldots, t'_m \in \mathcal{T}} \ \sum_{k=1}^{m} R(t'_k, C')\ \prod_{i=1}^{m} P(t'_i|t_i) \\
&= \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} R(t'_k, C')\ P(t'_k|t_k) \sum_{T'_{-k} \in \mathcal{T}^{m-1}} \prod_{i \neq k} P(t'_i|t_i) \\
&= \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} R(t'_k, C')\ P(t'_k|t_k) \\
&= \sum_{k=1}^{m} \widetilde{V}(t_k, C')
\end{aligned}$$

where $T'_{-k} = (t'_1, \ldots, t'_{k-1}, t'_{k+1}, \ldots, t'_m)$. The second equality is obtained using (3.2) and (3.4). The fourth equality follows from

$$\sum_{T'_{-k} \in \mathcal{T}^{m-1}} \prod_{i \neq k} P(t'_i | t_i) \; = \sum_{T'_{-k} \in \mathcal{T}^{m-1}} P(T'_{-k} | T_{-k}) = 1 \; .$$

A.2 Decomposition of value function of pair of cameras in max-plus algorithm

The proof of (4.8) follows from Section A.1, except the joint states of cameras in Section A.1 is replaced by states of pair of cameras. The local value function V_{ij} in (4.7) for a pair of cameras i and j is given by

$$\begin{aligned}
V_{ij}((T, c_i, c_j), a_i, a_j) &= \sum_{T' \in \mathcal{T}^m} R_{ij}((T', c'_i, c'_j)) \; P(T'|T) \\
&= \sum_{t'_1 \in \mathcal{T}, \ldots, t'_m \in \mathcal{T}} \; \sum_{k=1}^{m} \widetilde{R}_{ij}(t'_k, c'_i, c'_j) \; \prod_{i=1}^{m} P(t'_i | t_i) \\
&= \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} \widetilde{R}_{ij}(t'_k, c'_i, c'_j) \; P(t'_k | t_k) \; \sum_{T'_{-k} \in \mathcal{T}^{m-1}} \prod_{i \neq k} P(t'_i | t_i) \\
&= \sum_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} \widetilde{R}_{ij}(t'_k, c'_i, c'_j) \; P(t'_k | t_k)
\end{aligned}$$

where $T'_{-k} = (t'_1, \ldots, t'_{k-1}, t'_{k+1}, \ldots, t'_m)$. The second equality is obtained using (4.3) and (4.4). The fourth equality follows from

$$\sum_{T'_{-k} \in \mathcal{T}^{m-1}} \prod_{i \neq k} P(t'_i | t_i) \; = \sum_{T'_{-k} \in \mathcal{T}^{m-1}} P(T'_{-k} | T_{-k}) = 1 \; .$$

A.3 Value function decomposition in MDP framework for FOMT problem

The value function V (5.6) is given by,

$$\begin{aligned}
V(T_{\mathcal{M}}, C, A) &= \sum_{T'_{\mathcal{M}} \in \mathcal{T}^m} R((T'_{\mathcal{M}}, C'))P(T'_{\mathcal{M}}|T_{\mathcal{M}}, C') \\
&= \sum_{T'_{\mathcal{M}} \in \mathcal{T}^m} \min_{k \in \mathcal{M}} t'_{o_k} P(T'_{\mathcal{M}}|T_{\mathcal{M}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y} \min_{k \in \mathcal{Y}} t'_{o_k} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \sum_{T'_{\overline{\mathcal{Y}}} \in \mathcal{T}^{m-y}} P(T'_{\overline{\mathcal{Y}}}|T_{\overline{\mathcal{Y}}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y} \min_{k \in \mathcal{Y}} t'_{o_k} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y \setminus \mathcal{T}^y_{C'}} \min_{k \in \mathcal{Y}} t'_{o_k} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} \min_{k \in \mathcal{Y}} t'_{o_k} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y \setminus \mathcal{T}^y_{C'}} t_{\text{min}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} (t_{\text{min}} + 1) P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y \setminus \mathcal{T}^y_{C'}} t_{\text{min}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} t_{\text{min}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \\
&\quad + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') \\
&= \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y} t_{\text{min}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C') + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} P(T'_{\mathcal{Y}}|T_{\mathcal{Y}}, C')
\end{aligned}$$

continued in next page...

$$
\begin{aligned}
&= t_{\min} \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y} P(T'_{\mathcal{Y}} | T_{\mathcal{Y}}, C') + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} P(T'_{\mathcal{Y}} | T_{\mathcal{Y}}, C') \\
&= t_{\min} + \sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}} P(T'_{\mathcal{Y}} | T_{\mathcal{Y}}, C') \\
&= t_{\min} + \prod_{k \in \mathcal{Y}} \sum_{t'_k \in \mathcal{T}_{C'}} P(t'_k | t_k, C') \ .
\end{aligned}
$$

The first and second equalities are due to (5.6) and (5.4), respectively. The third equality follows by partitioning $T_{\mathcal{M}}$ into $T_{\mathcal{Y}} \triangleq (t_k)_{k \in \mathcal{Y}}$ and $T_{\overline{\mathcal{Y}}} \triangleq (t_k)_{k \in \overline{\mathcal{Y}}}$ where $\mathcal{M} = \mathcal{Y} \cup \overline{\mathcal{Y}}$ and $y = |\mathcal{Y}|$. The fourth equality follows from the law of total probability: $\sum_{T'_{\overline{\mathcal{Y}}} \in \mathcal{T}^{m-y}} P(T'_{\overline{\mathcal{Y}}} | T_{\overline{\mathcal{Y}}}, C') = 1$. The fifth equality is due to $\mathcal{T}^y = (\mathcal{T}^y \backslash \mathcal{T}^y_{C'}) \cup \mathcal{T}^y_{C'}$ where $\mathcal{T}^y_{C'}$ denotes the set of joint states of the y targets in $\mathcal{Y}$ whose locations all lie within $fov(C')$, i.e., $\mathcal{T}^y_{C'} \triangleq \{T'_{\mathcal{Y}} \in \mathcal{T}^y \mid \forall t'_k \in T'_{\mathcal{Y}} \ t'_{l_k} \in fov(C')\}$. To obtain the sixth equality, for a given $T'_{\mathcal{Y}} = (t'_k)_{k \in \mathcal{Y}}$,

$$
\min_{k \in \mathcal{Y}} t'_{o_k} \quad = \quad \begin{cases} t_{\min} & \text{if } T'_{\mathcal{Y}} \in \mathcal{T}^y \backslash \mathcal{T}^y_{C'}, \\ \\ t_{\min} + 1 & \text{if } T'_{\mathcal{Y}} \in \mathcal{T}^y_{C'}; \end{cases}
$$

where $t_{\min}$ is a constant with respect to the expectations. The second last equality follows from the law of total probability: $\sum_{T'_{\mathcal{Y}} \in \mathcal{T}^y} P(T'_{\mathcal{Y}} | T_{\mathcal{Y}}, C') = 1$. The last equality is due to the conditional independence property in the transition model (5.1).

A.4 Observation model factorization in POMDP framework

$$\begin{aligned}
P(Z|S) &= P(Z|T, C) \\
&= P(z_1, z_2, \ldots, z_m | t_1, t_2, \ldots, t_m, C) \\
&= \prod_{k=1}^{m} P(z_k | t_k, C) \\
&= \prod_{k=1}^{m} P(z_k | t_{l_k}, C) \; .
\end{aligned}$$

The last two equalities are due to the conditional independence assumption in the observation model (Section 6.3.3).

A.5 Posterior belief decomposition in POMDP framework

$$
\begin{aligned}
B'(S') &= \eta\, P(Z|S') \sum_{S\in\mathcal{S}} P(S'|S,A)B(S) \\
&= \eta \prod_{k=1}^{m} P(z_k|t'_{l_k},C') \sum_{T\in\mathcal{T}^m} \sum_{C\in\mathcal{C}^n} \prod_{k=1}^{m} P(t'_k|t_k) \\
&\quad \prod_{i=1}^{n} \delta_{\tau(c_i,a_i)}(c'_i) \prod_{k=1}^{m} b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}_i}(c_i) \\
&= \eta \prod_{k=1}^{m} P(z_k|t'_{l_k},C') \left(\sum_{T\in\mathcal{T}^m} \prod_{k=1}^{m} P(t'_k|t_k) \prod_{k=1}^{m} b_k(t_k) \right) \\
&\quad \left(\sum_{C\in\mathcal{C}^n} \prod_{i=1}^{n} \delta_{\tau(c_i,a_i)}(c'_i) \prod_{i=1}^{n} \delta_{\hat{c}_i}(c_i) \right) \\
&= \eta \prod_{k=1}^{m} P(z_k|t'_{l_k},C') \prod_{k=1}^{m} \sum_{t_k\in\mathcal{T}} P(t'_k|t_k)b_k(t_k) \prod_{i=1}^{n} \delta_{\tau(\hat{c}_i,a_i)}(c'_i) \\
&= \sum_{C'\in\mathcal{C}^n} \prod_{k=1}^{m} \eta_k \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \\
&\quad \prod_{k=1}^{m} P(z_k|t'_{l_k},C') \sum_{t_k\in\mathcal{T}} P(t'_k|t_k)b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \\
&= \prod_{k=1}^{m} \eta_k P(z_k|t'_{l_k},C') \sum_{t_k\in\mathcal{T}} P(t'_k|t_k)b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \\
&= \prod_{k=1}^{m} b'_k(t'_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)\ .
\end{aligned}
$$

The first equality is due to (6.1). The second equality follows from (6.2), (6.3), and (6.4). The fifth equality follows from $\eta = \sum_{C'\in\mathcal{C}^n} \prod_{k=1}^{m} \eta_k \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)$ (see proof in Section A.8). The last equality is due to (6.6).

A.6 Reward function decomposition in POMDP framework

$$
\begin{aligned}
R(B) &= \sum_{S\in\mathcal{S}} R(S)B(S) \\
&= \sum_{(T,C)\in\mathcal{S}} R((T,C))B((T,C)) \\
&= \sum_{C\in\mathcal{C}^n}\sum_{T\in\mathcal{T}^m}\sum_{k=1}^{m}\widetilde{R}(t_k,C)\prod_{k=1}^{m} b_k(t_k)\prod_{i=1}^{n}\delta_{\hat{c}_i}(c_i) \\
&= \sum_{k=1}^{m}\sum_{t_k\in\mathcal{T}}\widetilde{R}(t_k,\widehat{C})b_k(t_k)\sum_{T_{-k}\in\mathcal{T}^{m-1}}\prod_{j\neq k} b_j(t_j) \\
&= \sum_{k=1}^{m}\sum_{t_k\in\mathcal{T}}\widetilde{R}(t_k,\widehat{C})b_k(t_k) \\
&= \sum_{k=1}^{m}\left(\sum_{t_k\in\mathcal{T}}\widetilde{R}(t_k,\widehat{C})b_k(t_k)\right) \\
&= \sum_{k=1}^{m}\widetilde{R}(b_k,\widehat{C})
\end{aligned}
$$

where $T_{-k} = (t_1,\ldots,t_{k-1},t_{k+1},\ldots,t_m)$. The third equality is due to (6.4) and (6.7). The fifth equality follows from our independence assumption similar to that in (6.4) and the law of total probability:

$$
\sum_{T_{-k}\in\mathcal{T}^{m-1}}\prod_{j\neq k} b_j(t_j) = \sum_{T_{-k}\in\mathcal{T}^{m-1}} P(T_{-k}) = 1\ .
$$

A.7 Value function decomposition in POMDP framework

$$
\begin{aligned}
V(B,A) &= \sum_{Z\in\mathcal{Z}} R(B')P(Z|B,A) \\
&= \sum_{Z\in\mathcal{Z}} \sum_{k=1}^{m} \widetilde{R}(b'_k,\widehat{C}') \prod_{j=1}^{m} P(z_j|b_j,\widehat{C}') \\
&= \sum_{k=1}^{m} \sum_{Z\in\mathcal{Z}} \widetilde{R}(b'_k,\widehat{C}') \prod_{j=1}^{m} P(z_j|b_j,\widehat{C}') \\
&= \sum_{k=1}^{m} \sum_{z_k\in\dot{\mathcal{Z}}} \widetilde{R}(b'_k,\widehat{C}')P(z_k|b_k,\widehat{C}') \sum_{Z_{-k}\in\dot{\mathcal{Z}}^{m-1}} \prod_{j\neq k} P(z_j|b_j,\widehat{C}') \\
&= \sum_{k=1}^{m} \sum_{z_k\in fov(\widehat{C}')} \widetilde{R}(b'_k,\widehat{C}')P(z_k|b_k,\widehat{C}') \\
&= \sum_{k=1}^{m} \sum_{z_k\in fov(\widehat{C}')} \sum_{t'_k\in\mathcal{T}} \widetilde{R}(t'_k,\widehat{C}')b'_k(t'_k)P(z_k|b_k,\widehat{C}') \\
&= \sum_{k=1}^{m} \sum_{z_k\in fov(\widehat{C}')} \sum_{t'_k\in\mathcal{T}} \widetilde{R}(t'_k,\widehat{C}')\hat{b}'_k(t'_k) \\
&= \sum_{k=1}^{m} \widetilde{V}(b_k,\widehat{C}') = \sum_{k=1}^{m} \widetilde{V}(b_k,(\tau(\hat{c}_1,a_1),\ldots,\tau(\hat{c}_n,a_n)))
\end{aligned}
$$

where $\widehat{C}' \triangleq (\hat{c}'_1,\ldots,\hat{c}'_n)$ and $Z_{-k} = (z_1,\ldots,z_{k-1},z_{k+1},\ldots,z_m)$. The first equality is due to (6.11). The second equality is obtained using (6.8) and $\eta^{-1} = \sum_{C'\in\mathcal{C}^n} \prod_{k=1}^{m} \eta_k^{-1} \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)$ (Section A.8). The fifth equality follows from $P(Z_{-k}|B_{-k},A) = \prod_{j\neq k} P(z_j|b_j,\widehat{C}')$ where $B_{-k}(S) = \prod_{j\neq k} b_j(t_j) \prod_{i=1}^{n} \delta_{\hat{c}_i}(c_i)$ and then the law of total probability: $\sum_{Z_{-k}\in\dot{\mathcal{Z}}^{m-1}} \prod_{j\neq k} P(z_j|b_j,\widehat{C}') = 1$. Also, note that when $z_k \notin fov(\widehat{C}')$, $\widetilde{R}(b'_k,\widehat{C}') = 0$. The sixth equality is due to (6.9). Since the normalizing constant of $b'_k(t'_k)$ is $1/P(z_k|b_k,\widehat{C}')$, the seventh equality follows.

A.8 Derivation of η in POMDP framework

The proof of $\eta = \sum_{C' \in \mathcal{C}^n} \prod_{k=1}^{m} \eta_k \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)$ is as follows:

$$
\begin{aligned}
\eta^{-1} &= P(Z|B, A) \\
&= \sum_{S' \in S} P(Z|S') P(S'|B, A) \\
&= \sum_{S' \in S} P(Z|S') \sum_{S \in \mathcal{S}} P(S'|S, A) P(S|B) \\
&= \sum_{S' \in \mathcal{S}} P(Z|S') \sum_{S \in \mathcal{S}} P(S'|S, A) B(S) \\
&= \sum_{C' \in \mathcal{C}^n} \sum_{T' \in \mathcal{T}^m} \prod_{k=1}^{m} P(z_k|t'_{l_k}, C') \\
&\quad \sum_{C \in \mathcal{C}^n} \sum_{T \in \mathcal{T}^m} \left(\prod_{k=1}^{m} P(t'_k|t_k) \prod_{i=1}^{n} \delta_{\tau(c_i, a_i)}(c'_i) \right) \left(\prod_{k=1}^{m} b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}_i(c_i)} \right) \\
&= \sum_{C' \in \mathcal{C}^n} \sum_{T' \in \mathcal{T}^m} \prod_{k=1}^{m} P(z_k|t'_{l_k}, C') \left(\sum_{T \in \mathcal{T}^m} \prod_{k=1}^{m} P(t'_k|t_k) \prod_{k=1}^{m} b_k(t_k) \right) \\
&\quad \left(\sum_{C \in \mathcal{C}^n} \prod_{i=1}^{n} \delta_{\tau(c_i, a_i)}(c'_i) \prod_{i=1}^{n} \delta_{\hat{c}_i}(c_i) \right) \\
&= \sum_{C' \in \mathcal{C}^n} \sum_{T' \in \mathcal{T}^m} \prod_{k=1}^{m} P(z_k|t'_{l_k}, C') \prod_{k=1}^{m} \sum_{t_k \in \mathcal{T}} P(t'_k|t_k) b_k(t_k) \prod_{i=1}^{n} \delta_{\tau(\hat{c}_i, a_i)}(c'_i) \\
&= \sum_{C' \in \mathcal{C}^n} \sum_{T' \in \mathcal{T}^m} \prod_{k=1}^{m} P(z_k|t'_{l_k}, C') \prod_{k=1}^{m} \sum_{t_k \in \mathcal{T}} P(t'_k|t_k) b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \\
&= \sum_{C' \in \mathcal{C}^n} \prod_{k=1}^{m} \sum_{t'_k \in \mathcal{T}} P(z_k|t'_{l_k}, C') \sum_{t_k \in \mathcal{T}} P(t'_k|t_k) b_k(t_k) \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i) \\
&= \sum_{C' \in \mathcal{C}^n} \prod_{k=1}^{m} \eta_k^{-1} \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)
\end{aligned}
$$

where $\eta_k^{-1} = \sum_{t'_k \in \mathcal{T}} P(z_k|t'_{l_k}, C') \sum_{t_k \in \mathcal{T}} P(t'_k|t_k) b_k(t_k) = P(z_k|b_k, C')$. The fourth equality follows from (6.2), (6.3), and (6.4). It follows that $\eta = \sum_{C' \in \mathcal{C}^n} \prod_{k=1}^{m} \eta_k \prod_{i=1}^{n} \delta_{\hat{c}'_i}(c'_i)$.

Bibliography

[Ahmedali and Clark, 2006] T. Ahmedali and J. J. Clark. Collaborative Multi-Camera surveillance with automated person detection. In *Proc. Canadian Conference on Computer and Robot Vision*, page 39, 2006.

[Alahi *et al.*, 2008] A. Alahi, P. Vandergheynst, M. Bierlaire, and M. Kunt. Object detection and matching in a mixed network of fixed and mobile cameras. In *Proc. ACM workshop on Analysis and retrieval of events/actions and workflows in video streams*, pages 9–16, 2008.

[Arsic *et al.*, 2008] D. Arsic, E. Hristov, N. Lehment, B. Hornler, B. Schuller, and G. Rigoll. Applying multi layer homography for multi camera person tracking. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–9, 2008.

[Axi, 2011] Axis Communications., Axis 232D+ Network Dome Camera datasheet. [Available online] www.axis.com, 2011.

[Banerjee *et al.*, 2010] S. Banerjee, A. Chowdhury, and S. Ghosh. Video surveillance with ptz cameras: The problem of maximizing effective monitoring time. In *Distributed Computing and Networking*, volume 5935, pages 341–352. 2010.

[Bramberger *et al.*, 2005] M. Bramberger, M. Quaritsch, T. Winkler, B. Rinner, and H. Schwabach. Integrating multi-camera tracking into a dynamic task allocation system for smart cameras. In *Proc. Advanced Video and Signal Based Surveillance*, pages 474–479, 2005.

[Bruce and Gordon, 2004] Allison Bruce and Geoffrey Gordon. Better motion prediction for people-tracking. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA), Barcelona, Spain*, 2004.

[Bustamante *et al.*, 2012] A. L. Bustamante, J. M. Molina, and M. A. Patricio. Distributed active-camera control architecture based on multi-agent systems. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*, volume 156, pages 103–112. 2012.

[Bustamante *et al.*, 2013] A. L. Bustamante, J. M. Molina, and M. A. Patricio. A practical approach for active camera coordination based on a fusion-driven multi-agent system. *International Journal of Systems Science*, pages 1–15, 2013.

[Castanedo *et al.*, 2006] F. Castanedo, M. A. Patricio, J. Garca, and J. M. Molina. Extending surveillance systems capabilities using BDI cooperative sensor agents. In *Proc. ACM international workshop on Video surveillance and sensor networks*, pages 131–138, 2006.

[Cavallaro, 2005] A. Cavallaro. Event detection in underground stations using multiple heterogeneous surveillance cameras. In *Advances in Visual Computing*, pages 535–542. 2005.

[Chang and Gong, 2001] T. H. Chang and S. Gong. Tracking multiple people with a Multi-Camera system. In *Proc. Workshop on Multi-Object Tracking*, pages 19–26, 2001.

[Collins *et al.*, 2001] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proc. IEEE*, 89(10):1456–1477, 2001.

[Costello and Wang, 2005] C. J. Costello and I. J. Wang. Surveillance camera coordination through distributed scheduling. In *Proc. Conference on Decision and Control*, pages 1485–1490, 2005.

[Crick and Pfeffer, 2003] C. Crick and A. Pfeffer. Loopy belief propagation as a basis for communication in sensor networks. In *Proc. Conference on Uncertainty in Artificial Intelligence*, pages 159–166, 2003.

[Dieber *et al.*, 2011] B. Dieber, C. Micheloni, and B. Rinner. Resource-aware coverage and task assignment in visual sensor networks. *IEEE Trans. on Circuits and Systems for Video Technology*, 21(10):1424–1437, 2011.

[Ding *et al.*, 2012a] C. Ding, A. A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury. Opportunistic sensing in a distributed ptz camera network. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–6, 2012.

[Ding *et al.*, 2012b] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury. Collaborative sensing in a distributed ptz camera network. *IEEE Trans. on Image Processing*, 21(7):3282–3295, 2012.

[Duchi *et al.*, 2007] D. Duchi, J. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. In *Proc. Advances in Neural Information Processing Systems*, volume 19, page 369, 2007.

[El-Alfy *et al.*, 2009] H. El-Alfy, D. Jacobs, and L. Davis. Assigning cameras to subjects in video surveillance systems. In *Proc. International Conference on Robotics and Automation*, pages 837 –843, 2009.

[Fiore *et al.*, 2008] L. Fiore, D. Fehr, R. Bodor, A. Drenner, G. Somasundaram, and N. Papanikolopoulos. Multi-Camera human activity monitoring. *Intelligent and Robotics Syst.*, 52(1):5–43, 2008.

[Fleck *et al.*, 2006] S. Fleck, F. Busch, P. Biber, and W. Straber. 3D surveillance a distributed network of smart cameras for Real-Time tracking and its visualization in 3D. In *Proc. Computer Vision and Pattern Recognition Workshop*, page 118, 2006.

[Freeman and Liu, 2011] W. T. Freeman and C. Liu. Markov random fields for super-resolution and texture synthesis. *Advances in Markov Random Fields for Vision and Image Processing*, 2011.

[Fukuda *et al.*, 2000] T. Fukuda, T. Suzuki, F. Kobayashi, F. Arai, Y. Hasegawa, and M. Negi. Seamless tracking system with multiple cameras. In *Proc. Conference of the IEEE Industrial Electronics Society*, volume 2, pages 1249–1254, 2000.

[Garcia *et al.*, 2005] J. Garcia, J. Carbo, and J. M. Molina. Agent-based coordination of cameras. *International Journal of Computer Science and Applications*, 2(1):33–37, 2005.

[GE, 2009] GE Research. Website, 2009. http://tinyurl.com/gesurveillance.

[Gerkey *et al.*, 2003] B. P. Gerkey, R. T. Vaughan, and A. Howard. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proc. ICAR*, pages 317–323, 2003.

[Guestrin *et al.*, 2002] C. Guestrin, M. G. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *Proc. International Conference on Machine Learning*, pages 227–234, 2002.

[Guler *et al.*, 2003] S. Guler, J. M. Griffith, and I. A. Pushee. Tracking and handoff between multiple perspective camera views. In *Proc. Applied Imagery Pattern Recognition Workshop*, pages 275–281, 2003.

[Hampapur *et al.*, 2003] A. Hampapur, S. Pankanti, A. Senior, Ying-Li Tian, L. Brown, and R. Bolle. Face cataloger: multi-scale imaging for relating identity to location. In *Proc. International Conference on Advanced Video and Signal-Based Surveillance*, pages 13–20, 2003.

[Hayet *et al.*, 2005] J.B. Hayet, T. Mathes, J. Czyz, J. Piater, J. Verly, and B. Macq. A modular multi-camera framework for team sports tracking. In *Advanced Video and Signal Based Surveillance. IEEE Conference on*, pages 493–498, 2005.

[Hightower and Borriello, 2004] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *UbiComp 2004: Ubiquitous Computing*, pages 88–106. 2004.

[Hodge and Kamel, 2003] L. Hodge and M. Kamel. An agent-based approach to multisensor coordination. *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(5):648–661, 2003.

[Hu *et al.*, 2006] W. Hu, M. Hu, X. Zhou, and J. Lou. Principal Axis-Based correspondence between multiple cameras for people tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):663–671, 2006.

[Huang and Fu, 2011] C. M. Huang and L. C. Fu. Multitarget visual tracking based effective surveillance with cooperation of multiple active cameras. *IEEE Trans. SMC, Part B: Cybernetics*, 41(1):234 –247, 2011.

[IBM, 2012] IBM Smart Surveillance System (S3). Website, 2012. http://researcher.watson.ibm.com/researcher/view_project.php?id=1903.

[Ilie and Welch, 2011] A. Ilie and G. Welch. On-line control of active camera networks for computer vision tasks. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–6, 2011.

[Jain *et al.*, 1984] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.

[Jones and Mitter, 2006] P. B. Jones and S. K. Mitter. An iterative algorithm for autonomous tasking in sensor networks. In *Proc. Decision and Control*, pages 2740–2746, 2006.

[Kerhet *et al.*, 2007] A. Kerhet, F. Leonardi, A. Boni, P. Lombardo, M. Magno, and L. Benini. Distributed video surveillance using hardware-friendly sparse large margin classifiers. In *Proc. Advanced Video and Signal Based Surveillance*, pages 87–92, 2007.

[Khoshabeh *et al.*, 2007] R. Khoshabeh, T. Gandhi, and M. M. Trivedi. Multi-camera based traffic flow characterization & classification. In *Proc. Intelligent Transportation Systems Conference*, pages 259–264, 2007.

[Ko and Berry, 2005] T. H. Ko and N. M. Berry. Distributed calibration and tracking with low-power image sensors. In *Proc. Conference on Diversity in Computing*, pages 40–43, 2005.

[Kok and Vlassis, 2006] J. R. Kok and N. Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In *RoboCup 2005*, pages 1–12. 2006.

[Krahnstoever *et al.*, 2008] N. Krahnstoever, T. Yu, S. N. Lim, K. Patwardhan, and P. Tu. Collaborative Real-Time Control of Active Cameras in Large Scale Surveillance Systems. In *Proc. ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[Krumm *et al.*, 2000] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-Camera Multi-Person tracking for EasyLiving. In *Proc. International Workshop on Visual Surveillance*, pages 3–10, 2000.

[Kuyer *et al.*, 2008] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis. Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Proc. of the European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 656–671, 2008.

[Lan *et al.*, 2006] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order markov random fields. In *Proc. European Conference on Computer Vision - Volume Part II*, pages 269–282, 2006.

[Lee *et al.*, 2007] D. S. Lee, C. M. Chen, and C. Y. Tang. Weighted fair queueing and compensation techniques for wireless packet switched networks. *IEEE Trans. on Vehicular Technology*, 56(1):297–311, 2007.

[Li and Bhanu, 2011] Y. Li and B. Bhanu. Utility-based camera assignment in a video network: A game theoretic framework. *IEEE Sensors Journal*, 11(3):676 –687, 2011.

[Liu and Leung, 2008] E. Liu and K.K. Leung. Proportional fair scheduling: Analytical insight under rayleigh fading environment. In *Proc. Wireless Communications and Networking Conference*, pages 1883–1888, 2008.

[Liu *et al.*, 2002] Q. Liu, D. Kimber, J. Foote, L. Wilcox, and J. Boreczky. FlySPEC: a multiuser video camera system with hybrid human and automatic control. In *Proc. International Conference on Multimedia*, pages 484–492, 2002.

[Lu and Payandeh, 2008] Y. Lu and S. Payandeh. Cooperative hybrid multi-camera tracking for people surveillance. In *Proc. Canadian Conference on Electrical and Computer Engineering*, pages 1365–1368, 2008.

[Luber *et al.*, 2010] Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 464–469. IEEE, 2010.

[Matsuyama and Ukita, 2002] T. Matsuyama and N. Ukita. Real-time multitarget tracking by a cooperative distributed vision system. *Proc. of the IEEE*, 90(7):1136–1150, 2002.

[Medina and Benekohal, 2012] J. C. Medina and R. F. Benekohal. Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy. In *Proc. International Conference on Intelligent Transportation Systems*, pages 596–601, 2012.

[MER, 2009] Mitsubishi Electric Research Laboratories. Website, 2009. http://www.merl.com/areas/VideoMining/.

[Micheloni *et al.*, 2005] C. Micheloni, E. Salvador, F. Bigaran, and G. L. Foresti. An integrated surveillance system for outdoor security. In *Proc. Advanced Video and Signal Based Surveillance*, pages 480–485, 2005.

[Micheloni *et al.*, 2010] C. Micheloni, B. Rinner, and G.L. Foresti. Video analysis in pan-tilt-zoom camera networks. *IEEE Signal Processing Magazine*, 27(5):78–90, 2010.

[Mittal and Davis, 2001] A. Mittal and L. Davis. Unified multi-camera detection and tracking using region-matching. In *Proc. IEEE Workshop on Multi-Object Tracking*, page 3, 2001.

[Morye *et al.*, 2013] A. A. Morye, C. Ding, A. K. Roy-Chowdhury, and J. A. Farrell. Constrained optimization for opportunistic distributed visual sensing. In *American Controls Conference*, 2013.

[Muller and Anido, 2004] J. B. Muller and R. O. Anido. Distributed real-time soccer tracking. In *Proc. International workshop on Video surveillance and sensor networks*, pages 97–103, 2004.

[Muoz-Salinas *et al.*, 2009a] R. Muoz-Salinas, R. Medina-Carnicer, F. J. Madrid-Cuevas, and A. Carmona-Poyato. Multi-camera people tracking using evidential filters. *International Journal of Approximate Reasoning*, 50(5):732–749, 2009.

[Muoz-Salinas *et al.*, 2009b] R. Muoz-Salinas, R. Medina-Carnicer, F. J. Madrid-Cuevas, and A. Carmona-Poyato. People detection and tracking with multiple stereo cameras using particle filters. *Visual Communication and Image Representation*, 20(5):339–350, 2009.

[Nandhakumar and Aggarwal, 1988] N. Nandhakumar and J. K. Aggarwal. Integrated analysis of thermal and visual images for scene interpretation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4):469–481, 1988.

[Nandhakumar, 1992] N. Nandhakumar. Robust feature evaluation for multisensory computer vision. In *Proc. Pattern Recognition Conference B: Pattern Recognition Methodology and Systems, Proceedings.*, pages 446–449, 1992.

[Natarajan *et al.*, 2012a] P. Natarajan, T. N. Hoang, K. H. Low, and M. Kankanhalli. Decision-theoretic approach to maximizing observation of multiple targets in multi-camera surveillance. In *Proc. International Conference on Autonomous Agents and MultiAgent Systems*, pages 155–162, 2012.

[Natarajan *et al.*, 2012b] P. Natarajan, T.N. Hoang, K.H. Low, and M. Kankanhalli. Decision-theoretic coordination and control for active multi-camera surveillance in uncertain, partially observable environments. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–6, 2012.

[Natarajan, 2012a] P. Natarajan. Decision-theoretic approach for controlling and coordinating multiple active cameras in surveillance. In *Proc. International Conference on Autonomous Agents and MultiAgent Systems [Doctoral consortium abstract]*, 2012.

[Natarajan, 2012b] P. Natarajan. Phd forum: Decision-theoretic coordination and control for active multi-camera surveillance. In *Proc. International Conference on Distributed Smart Cameras [PhD Forum]*, 2012.

[Paek *et al.*, 2007] E. Paek, C. Park, M. Ki, K. Park, and J. Paik. Mutiple-view object tracking using metadata. In *Proc. International Conference on Wavelet Analysis and Pattern Recognition*, volume 1, pages 12–17, 2007.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, 1988.

[pet, 2013] Dataset - PETS: Performance Evaluation of Tracking and Surveillance, 2013.

[Petrushin *et al.*, 2006] V. A. Petrushin, G. Wei, and A. V. Gershman. Multiple-camera people localization in an indoor environment. *Knowledge and Information System*, 10(2):229–241, 2006.

[Piciarelli *et al.*, 2009] C. Piciarelli, C. Micheloni, and G. L Foresti. PTZ camera network reconfiguration. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–7, 2009.

[Prati *et al.*, 2005] A. Prati, R. Vezzani, L. Benini, E. Farella, and P. Zappi. An integrated multimodal sensor network for video surveillance. In *Proc. International workshop on Video surveillance and sensor networks*, pages 95–102, 2005.

[Qureshi and Terzopoulos, 2005] F. Z. Qureshi and D. Terzopoulos. Surveillance camera scheduling: a virtual vision approach. In *Proc. International Workshop on Video surveillance and sensor networks*, pages 131–140, 2005.

[Qureshi and Terzopoulos, 2008] F. Qureshi and D. Terzopoulos. Smart camera networks in virtual reality. *Proc. of the IEEE*, 96(10):1640–1656, 2008.

[Qureshi and Terzopoulos, 2009] F. Z. Qureshi and D. Terzopoulos. Planning ahead for PTZ camera assignment and handoff. In *Proc. International Conference on Distributed Smart Cameras*, 2009.

[Qureshi, 2010] F. Z. Qureshi. Collaborative sensing via local negotiations in ad hoc networks of smart cameras. In *Proc. International Conference on Distributed Smart Cameras*, pages 190–197, 2010.

[Raz *et al.*, 2004] D. Raz, H. Levy, and B. Avi-Itzhak. A resource-allocation queueing fairness measure. *SIGMETRICS Perform. Eval. Rev.*, 32(1):130–141, 2004.

[Rinner and Wolf, 2008] B. Rinner and W. Wolf. A bright future for distributed smart cameras. *Proc. of the IEEE*, 96(10):1562–1564, 2008.

[Scheunert *et al.*, 2004] U. Scheunert, H. Cramer, B. Fardi, and G. Wanielik. Multi sensor based tracking of pedestrians: a survey of suitable movement models. In *Proc. Intelligent Vehicles Symposium*, pages 774–778, 2004.

[Singh *et al.*, 2008] V. K. Singh, P. K. Atrey, and M. Kankanhalli. Coopetitive multi-camera surveillance using model predictive control. *Machion Vision Application*, 2008.

[Sommerlade and Reid, 2010] E. Sommerlade and I. Reid. Probabilistic surveillance with multiple active cameras. In *Proc. International Conference on Robotics and Automation*, 2010.

[Song *et al.*, 2008a] B. Song, C. Soto, A. K. Roy-Chowdhury, and J. A. Farrell. Decentralized camera network control using game theory. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–8, 2008.

[Song *et al.*, 2008b] B. Song, C. Soto, A. K. Roy-Chowdhury, and J. A. Farrell. Decentralized camera network control using game theory. In *Proc. International Conference on Distributed Smart Cameras*, pages 1–8, 2008.

[Song *et al.*, 2011a] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-chowdhury. Distributed camera networks. *IEEE Signal Processing Magazine*, 28(3):20–31, 2011.

[Song *et al.*, 2011b] B. Song, C. Ding, A.T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury. Distributed camera networks. *IEEE Signal Processing Magazine*, 28(3):20–31, 2011.

[Song *et al.*, 2011c] B. Song, C. Ding, A. K. Roy-Chowdhury, and J. Farrell. Persistent observation of dynamic scenes in an active camera network. In *Distributed Video Sensor Networks*, pages 259–271. 2011.

[Soto *et al.*, 2009] C. Soto, B. Song, and A. K. Roy-Chowdhury. Distributed multi-target tracking in a self-configuring camera network. In *Proc. Computer Vision and Pattern Recognition*, 2009.

[Spaan and Lima, 2009] M. T. J. Spaan and P. U. Lima. A decision-theoretic approach to dynamic sensor selection in camera networks. In *Proc. International Conference on Automated Planning and Scheduling*, pages 279–304, 2009.

[Starzyk and Qureshi, 2011a] W. Starzyk and F. Z. Qureshi. Learning proactive control strategies for ptz cameras. In *Proc. International Conference on Distributed Smart Cameras*, pages 1 –6, 2011.

[Starzyk and Qureshi, 2011b] W. Starzyk and F. Z. Qureshi. Multi-tasking smart cameras for intelligent video surveillance systems. In *Proc. International Conference on Advanced Video and Signal-Based Surveillance*, pages 154–159, 2011.

[Tassiulas and Sarkar, 2002] L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless networks. In *Proc. INFOCOM*, volume 2, pages 763–772, 2002.

[Tastan, 2013] Bulent Tastan. *Learning Human Motion Models* . PhD thesis, University of Central Florida Orlando, Florida, 2013.

[Thrun *et al.*, 2005] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[Tsai, 1986] R. Y. Tsai. An efficient and accurate camera calibration technique for 3D machine vision. In *Proc. Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, FL., 1986.

[Viola and Jones, 2004] P. Viola and M. J. Jones. Robust real-time face detection. *Computer Vision*, 57:137–154, 2004.

[Vlassis *et al.*, 2004] N. Vlassis, R. Elhorst, and J. R. Kok. Anytime algorithms for multiagent decision making using coordination graphs. In *Proc. Systems, Man and Cybernetics*, volume 1, pages 953–957 vol.1, 2004.

[Wainwright *et al.*, 2004] M. Wainwright, T. Jaakkola, and A. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing*, 14(2):143–166, 2004.

[Wang *et al.*, 2011] Y. Wang, P. Natarajan, and M. Kankanhalli. Multi-camera skype: Enhancing the quality of experience of video conferencing. In *Proc. Pacific-Rim Conference on Multimedia*, 2011.

[Wang, 2003] J. Wang. Experiential sampling for video surveillance. In *Proc. Int. Workshop on Video Surveillance*, pages 77–86, 2003.

[Ward and Naish, 2009a] C. Ward and M.D. Naish. Scheduling active camera resources for multiple moving targets. In *Proc. Canadian Conference on Electrical and Computer Engineering*, pages 528–532, 2009.

[Ward and Naish, 2009b] C. Ward and M.D. Naish. Scheduling active camera resources for multiple moving targets. In *Proc. Canadian Conference on Electrical and Computer Engineering*, pages 528 –532, 2009.

[Yuan *et al.*, 2003] X. Yuan, Z. Sun, Y. Varol, and G. Bebis. A distributed visual surveillance system. In *Proc. Advanced Video and Signal Based Surveillance*, pages 199–204, 2003.

[Zongjie and Bhattacharya, 2011] T. Zongjie and P. Bhattacharya. A game-theoretic design for collaborative tracking in a video camera network. In *Proc. Advanced Video and Signal-Based Surveillance*, pages 474–479, 2011.