UNIVERSITY OF CALIFORNIA
RIVERSIDE

Camera Selection, Handoff and Control in Video Networks

A Dissertation submitted in partial satisfaction

of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Yiming Li

June 2012

Dissertation Committee:

Dr. Bir Bhanu, Chairperson

Dr. Matthew J. Barth

Dr. Chinya V. Ravishankar

The Dissertation of Yiming Li is approved:

_____

_____

_____

Committee Chairperson

University of California, Riverside

Acknowledgements

Finishing this dissertation ends one major chapter of my life and triggers the beginning of another. As with any major step in life, I feel lucky and grateful for the endless support from many people. Without their support, I could not have reached this point.

First, I have to thank my advisor, Dr. Bir Bhanu, for his guidance and continuous support over the past five years. He gave me the freedom to explore my ideas and was always willing to work through and discuss them with me. He taught me how to choose a research topic, do research and write a high-quality paper. He exemplifies a distinguished scholar, a motivating advisor and a true friend. I also want to thank Dr. Matthew J. Barth and Dr. Chinya V. Ravishankar for serving on my dissertation committee. Their constructive suggestions helped to improve the quality of this dissertation.

I would also like to thank former and current members of the VISLAB at UCR, Rui Li, Le An, Xiaojing Chen, Zhixing Jin, Yanping Chen, Linan Feng, Songfan Yang, Yu Sun, Vincent Nguyen and Asong Tambo, for their help on both study and daily life during my stay at VISLAB.

In addition to the support from VISLAB, I also earned many helps from other friends, Bi Song, Li Yu, Qunfeng He, Fran Jeng, Yexiong Feng, Li Zhang, Huilin Xu and Yuanyuan Lei. It is these friendships making me never feel lonely during the days in the United States.

Last but certainly not the least, I dedicate my accomplishment to my parents and husband. They do not only give me endless support, but discuss with me all kinds of valuable ideas, which broaden my views in related areas. Without their selfless love and

support throughout my life, I will not be what I am today. I would like to express my special thanks to my husband, for what he has done for me, which will be deeply engraved on my heart.

ABSTRACT OF THE DISSERTATION

Camera Selection, Handoff and Control in Video Networks

by

Yiming Li

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2012
Dr. Bir Bhanu, Chairperson

Due to the broad coverage of an environment and the possibility of coordination among different cameras, video sensor networks have attracted much interest in recent years. Although the field-of-view (FOV) of a single camera is limited and cameras may have overlapping/non-overlapping FOVs, seamless tracking of moving objects is desired. As the increasing of the video network complexity, there are more and more camera nodes in a network. This makes it hard for human observers to take care of the entire system and brings the emergence of the camera selection, handoff and control technologies.

In this study, we introduce a series of economics frameworks into the camera selection, handoff and control problem. This starts with two game theoretical approaches – the potential game approach and the weakly acyclic game approach. With these two methods, we can model the camera selection and handoff problem as a multiplayer game. Existing learning algorithms in the game theory literature make it efficient to find an optimal as well as stable solution to this problem.

As camera selection and handoff largely depend on the accuracy of the applied trackers, we develop a technique to jointly consider the tracking problem and the camera selection problem. In this work, fusion of multiple trackers is integrated with the camera selection process in a closed-loop manner.

Finally, active camera controls are considered by using the auction protocol. Unlike previous work, the bid price is formulated to have a vector representation, such that when a camera is available to follow multiple objects, we consider the "willingness" of this camera to track a particular object. Meanwhile, the potentially available cameras can also be considered to follow an object after some panning or tilting operations. Most of the computation is decentralized by computing the bid price locally while the final assignment is made by a virtual auctioneer based on all the available bids, which is analogous to a real auction in economics. Thus, we can take the advantage of distributed/centralized computation and avoid their pitfalls.

All these approaches are evaluated with real-world data under the VideoWeb [50] camera network environment. These proposed approaches are also compared with each other and other approaches. The experimental results show the robustness and efficacy of this study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The growing demand for security at airports, train stations, protective installations, wild life, banks, shopping malls, homes, etc. makes video network an active research area to meet the needs for video surveillance and monitoring [21]. Because of the large number of sensor nodes, video sensor networks make many complicated surveillance tasks possible over a large geographical area. Significant applications of video network include object detection, tracking, recognition and activity analysis from multiple cameras. The cameras in a network can collaborate with one another and can, thus, perform various tasks in a cooperative manner.

In traditional video surveillance systems, there is usually a control center with a wall of monitors displaying videos from the camera nodes. Human observers are used to observe all these videos simultaneously. Digital matrix technologies allow the human observer to switch videos from different cameras to be displayed, as shown in Figure 1.1. However, as the number of camera nodes increases dramatically, it is hard for a human being to do all the necessary switches to follow all the objects in the system. Thus, the problem of automatic video surveillance has risen to the forefront of the video sensor networks. This requires three capabilities: camera selection, camera handoff, and active camera control, which are the three main subjects of this study. We define camera

Figure 1.1: The control room of nowadays video surveillance system.

selection as a camera-object map, which tells us at each time instant which camera is being used to follow which object. Camera handoff is a dynamic process that the system transfers the right of tracking an object from one camera to another without losing the object in the network. The availability of camera handoff capability will provide the much needed situation assessment of the environment under surveillance. If the camera is a PTZ (pan / tilt / zoom) camera, then active controls are also available. By panning or tilting a camera, we can achieve a larger area under monitored. The zoom-in operation can provide us a close-up view of an object. For example, when a person's frontal view is available, zooming in the camera to have a close-up view of the face may provide us more information of interest.

A large amount of work has been done in the field of multi-camera multi-person tracking, as shown in the **first part** of this study (**Chapter 2**). However, to our knowledge, most of the existing work does not focus on the optimal camera selection, especially when different criteria are applied to different cameras. Conventional methods can only hand over from one camera to another when the object is leaving the FOV of

one camera and entering the FOV of another [25], which means that even if an object can be tracked continuously by using some of the existing approaches, we cannot get the "best" view (defined by the user) of this person at all times. This may waste a lot of information of interest, such as the frontal view of a person, which may be originally available in the video network. On the contrary, in our system, we want to make sure that the selected camera(s) can provide the best information about the selected object(s). To make the whole system more computationally efficient, we achieve this by viewing the camera selection process as a multi-agent system. The individual cameras are autonomous and can only know limited information about other cameras; meanwhile, these cameras cooperate to achieve the best system performance. Competition, or conflict, also exists in the fact that every camera wants to win the right to track the object(s) to increase its own welfare. So, in the camera selection problem in our system, every camera, like all the rational players in a game, tries to minimize its own cost and maximize its welfare, while all the cameras have to act cooperatively to give a better system performance at the same time. This is analogous to a typical game theory problem, which provides a mathematical foundation to solve the problems that involve competitive and dynamic interactions among those participants.

In the **second part** of this study (**Chapter 3**), we first model the camera selection and handoff problem as a potential of game. The merit of our approach is that it is independent of the camera topology. When multiple cameras are used for tracking and where multiple cameras can "see" the same object, the algorithm can automatically provide an *optimal* as well as *stable* solution of the camera selection. Since game

theoretic approach allows dealing with multiple criteria optimization, we are able to choose the "best" camera based on multiple criteria that are selected *a priori*. The detailed camera calibration or 3D scene understanding is not needed in our approach.

After that, to avoid the requirement that the local utility and global utility have to be aligned with each other, in the **third part** of this study (**Chapter 4**), we introduce the weakly acyclic game model. Compared with the potential game approach, weakly acyclic game, which is a superclass of the potential game, covers a broader scope of games. It provides more flexibility since the camera utility does not have to be exactly aligned with the global utility as it is required in a potential game. So we do not consider the alignment of the camera utility with global utility in the proposed approach, which also makes it easier than the potential game approach to realize the distributed control. Due to this flexibility, we can have different criteria for different cameras in the network. Camera handoffs take place automatically according to the calculated camera action assignment.

Since camera selection is always done based on the given tracking results, it is reasonable to combine tracking and camera selection to get better results for both sides. In the **fourth part** of this study (**Chapter 5**), we develop an approach to jointly consider optimal tracking and camera selection. Most of the recent work [63][64][65][66] uses only homography to build up the connection between tracking in multiple cameras. Although the ground plane homography can provide correspondence of the object's location in different camera views, it is not enough to resume or re-initialize a tracker with this location only. There are two major problems for doing so: (1) We do not have a good estimate of the object size with knowing its location only. (2) The use of ground

plane homography is under the assumption that the objects are persons walking on the same g round plane and their feet are visible. In this study, we propose the idea that uses homography and epipolar geometry together to better locate an object's position. This information and the camera selection results are combined together to re-initialize inaccurate trackers, which improves the system's performance consistently. We also apply a feet test to provide this feedback with a confidence for the existence of feet in a bounding box. The purpose of this work is to provide a reliable track for each individual person in a medium density crowd. Applications of this work can be in banks, home and residential CCTV systems, schools, etc.

Finally, in the **fifth part** of this study (**Chapter 6**), active camera controls by using auction mechanisms are developed. Auction-based approaches are used in multi-agent systems (multi-robot systems, manufacturing systems) for resource/task allocation problems. In an auction-based scenario, there is an auctioneer auctioning a good and all the potential buyers calculate their bids for the good locally. Finally, the auctioneer decides whom to sell the good based on the buyers' bids. This process, to a large extent, distributes the heaviest load of computation, the computation for bids, to each buyer, while the final decision is still optimal as long as there is a reasonable mechanism to make all the buyers rational. In this study, we model the process of grouping cameras to follow multiple objects in a camera network as the process of an economic auction. There is a virtual auctioneer holding an auction for each object to be followed and all the potential cameras bidding for it. By doing so, we benefit from the auction mechanism for distributed computation and consider the "willingness" of buyers (cameras). We choose

from the top N bids to form a group and, thus, make the cameras with higher potentials to work collaboratively.

The overall study is concluded in **part six (Chapter 7)**. Future works are also discussed in part six.

Chapter 2

Literature Review and Contributions of This Thesis

2.1. Related Work

There have been many papers discussing approaches for doing camera selection, handoff and control in a video network. In this chapter, we review these works in three categories separately: 1) Research in camera selection and handoff and control. 2) Research in integrating object tracking with camera selection and control.

2.1.1. Research in Camera Selection, Handoff and Control

With the development of camera networks, the number of camera nodes is increasing rapidly. It is becoming more and more unrealistic to display all the camera images to track an object. What is more desirable is that a system chooses a camera with an optimal view and displays it. Recently, there have been many papers dedicated to automatic camera selection. For example, [74] used many geometrical constraints to predict the possible directions that an object may be going and selected cameras in those directions. [75] used homography and camera calibration to select best cameras for a tennis court. [76] projected the likelihood map of an ensemble tracker to the ground map and chose the camera with a higher confidence. [76] reasoned about the dependencies of occlusions and confusions on the presence of persons and yielded an order for the inference of each person in a group of people and a set of cameras. [2] constructed a

look-up table based on cameras' viewing frustums and, then, camera selection is done by calculating the overlap between the current camera's viewing frustum and that of the sending camera. [78] maximized the information utility from multiple cameras subject to the constraint on the average energy consumption and selected a subset of cameras. [8] used the number of detected foreground blocks in a camera and the angle between the camera and the detected object to decide when to hand off to another camera. [79] proposed an approach to do consistent labeling of the objects in a video network. A camera transition graph (CTG) is built. This approach considered camera hand-offs at the edges of FOVs only. Similarly, [25] proposed the fuzzy-based system where rules are applied for camera hand-offs when an object is leaving the FOV of a camera and entering the FOV of another camera. Most of these works used a single tracker only.

Overall, the research work in camera selection and handoff in a video network consisting of multiple cameras can be classified according to many different aspects, such as whether it is embedded/PC-based; distributed/centralized; calibration-needed/calibration-free; topology-based/topology-free; statistics-based/statistics-free; uses/does not use master-slave scheme, etc.

Embedded systems have limited resources, such as memory, computing performance and power. Thus, approaches designed for embedded systems have to consider these factors and only simple approaches have been applied to such systems [32][33][2]. Distributed [25][34][35][4] systems have low bandwidth requirement since there is no need to transfer raw images. It is easy for such systems to increase the number of nodes and it is hard for a distributed system to die fully. Calibration can provide the

topology of the camera network and it is a must when the zooming-in/zooming-out operation is needed. But most of the calibration process needs pre-processing and can be time consuming. Topology-based approaches [1][2][3][4] rely on the geometrical relationships among cameras. These relationships tend to become quite complicated when the topology becomes complex and it is difficult to learn the topology based on the random traffic patterns [5]. Statistical-based approaches [6][7][8][9][10] usually depend on the objects' trajectories, while other factors such as orientation, shape, face etc., which are also very important for visual surveillance, are not considered. Master-slave scheme [34][14][8][8] uses a master (or a principal) camera to get the dominant view of an object while the slave cameras (or the helper cameras) cooperate with the master camera to keep the complete track of an object. Approaches that fall into this category mostly focus on camera selection with no or limited active control. Images in 3D are generated in some systems, such as [33]. However, in most approaches for the camera selection and handoff, only 2D images are deployed. There are also some approaches that have other considerations, such as resource allocation [36], fusion of different types of sensors [37][38], etc.

Some recent work [39][40] considers the camera selection problem in a more systematic way. However, all these above approaches do not consider the "best camera" selection. It is important to select the "best camera" because by doing so we can minimize the number of camera in a network to perform a given task, and therefore, for the same number of cameras, we can free up more cameras to carry out other important tasks. For example, in the case of pedestrian tracking, if we can obtain a person's face

(which contains the much needed information on many occasions) whenever it's available, we can obtain more interesting information that can help to recognize a person. In this study, we focus on how to do camera selection and handoff based on user-supplied criteria to make sure at each time instant the "best" camera is used to track a particular object.

Auction-based technique shows its effectiveness in solving many problems in multi-agent systems. For example, auction-based mechanism is established in [102] by He and Ioerger for computational grids. Gerkey and Mataric [103] use the auction method for dynamic task allocation for groups of failure-prone autonomous robots. Dias and Stentz [104] propose an opportunistic optimization approach for auction-based multirobot control. Leaders are used to do optimization within subgroups. Chen et al. [105] achieve single target tracking in wireless networks by deploying auction-based coalition. However, there is no work has ever used the auction-based technique in a camera network to select cameras to follow up multiple objects. In this study, we will apply auction-based mechanism to do camera active control. By using this approach, we can pre-estimate the best (potentially) available camera and make decisions to pan / tilt a camera or not.

2.1.2. Research in Integrating Object Tracking with Camera Selection, Handoff and Control

Since the proposed camera selection, handoff and control approaches largely depend on the trackers used, in this category, we review the state-of-the-art tracking algorithms and the integration of tracking algorithms with camera selection scheme works. We

categorize the related work into four areas: (1) Fusion of multiple trackers; (2) Multi-camera tracking; (3) Camera selection; and (4) Combination of multi-camera tracking and camera selection.

(1) Most of the fusion techniques for multiple trackers apply to single camera only. For example, [67] proposed an algorithm for fusion of multiple trackers. This approach used a classifier to determine if multiple trackers' results agree with each other. However, it required using synthetic data to do training, which is not desired for real-life video surveillance systems. [68] proposed an approach to locate an object's next position by doing a weakly supervised learning. This work can learn online from multiple imperfect. However, this approach is considered only under the single camera scenario. [69] proposed a framework for combing multiple trackers. Only kernel-based trackers and CONDENSATION-based trackers are considered in this work. [70] utilized multiple "tracking modules" (such as motion detector, region tracker, head detector and active shape tracker), to insure the tracking results. However, these modules are different steps for any tracking systems; this approach did not actually "fuse" any trackers. The fusion process of all the above approaches considered only the trackers' accuracy and did not take into account the object's appearance quality in multiple cameras. None of these approaches did the camera selection simultaneously with the fusion of multiple trackers.

(2) In the second category, many of the multi-camera tracking approaches used the homography constraint to build up correspondences among different cameras with the help of the ground plane homography. In these works, the key assumption is that all the persons walked on the ground plane that can be seen by all the cameras. Meanwhile, the

persons' feet are visible so that the homography can transform a person's position from one camera to another. Under this assumption, [66] used region covariance-based PSO tracking and built correspondences between two cameras using the ground plane homography. When an object is occluded in one camera, the information from another camera together with the homography is used to resume the tracker in the previous camera. [65] did a similar work, where a 5D (position, velocity and intensity) particle filter tracker is proposed and the position information is fused from multiple cameras using homography. [64] work is slightly different. Instead of aligning the feet of the person, the authors in this chapter aligned heads of persons from different views. The assumption is that only head from the same person will be aligned from different views. [69] proposed an online boosting system that used the homography information to match views from one camera to another, where they applied non-maximum suppression and non-minimum suppression to do the co-training. [72] proposed a context-base tracker switching approach. In this approach, multiple cameras are used to generate the 3D location of an object. The radiuses of the inner and outer "sphere of influence" highly depended on the camera calibration accuracy. Similarly, [73] uses multiple overlapping cameras to predict an object's 3D location, rejected the single intersections and kept the multiple-time intersections as the object's true location in the world coordinates. In all of these works, no camera hand-off or selection took place.

(3) For related works in camera selection and handoff, please refer to the previous Chapter 2.

In the above three categories, multi-camera tracking and camera selection are

treated separately. However, due to the dependencies on one another, it is natural to combine these two tasks together. In this fourth category, very little work has been done. The work that is most similar to this chapter is done by [76]. They do data fusion of multiple sensors, including IR cameras and integrate the fused results with a Kalman filter. Camera selection is done based on the appearance ratio criterion. Unlike this work, in this study, we propose an approach that is independent of the trackers that are used.

2.2. Contributions of This Thesis

Our study differs from the conventional approaches discussed in Section 2.1, in the following key aspects:

1. ***Game Theoretic Approach:*** We propose a series of game theoretic approaches for camera selection, handoff and control problem using the potential game (vehicle-target) model [26], the weakly acyclic game model [27] and the auction protocol [28], respectively. By using these models, we allow for both coordination and conflicts among the cameras.

2. ***Multiple Criteria for Tracking***: Multiple criteria are used in the design of *utility (payoff)* functions for the objects being tracked. The equilibrium of the game provides the solution of the camera selection. The bargaining mechanism and the payoff-based learning make sure that we can get a stable solution, which is optimal or near optimal, after only a small number of iterations.

3. ***"Best" Camera Selection***: We do not use the traditional master-slave system. Instead, by selecting the "best" camera(s), we can have a good enough view,

based on the user-supplied criteria, for observation of some specific target and simultaneously free the other cameras in the network for other tasks. Thus, the system can perform the tracking task with a minimum number of cameras, or, can perform more tasks with the same number of cameras.

4. ***Camera Active Control***: We use pre-calculated homographies plus the epipolar geometry to make the cameras "think" ahead, so that camera active control is also available based on user-supplied criteria. Potentially available cameras are also taken into account, which is very different from the existing works [28].

5. ***Fusion of Multiple Trackers***: We propose a score-level fusion of multiple trackers [29]. This framework is a black box of the tracker used, make it easy to integrate new tracker into the same framework. During the process of fusion, camera selection is jointly considered. Trackers with poor camera selection scores are given penalties.

6. ***Experimental Results***: Unlike some of the previous work [39], we evaluate the proposed approach in the context of real network using real data and show promising results with comparison to many state-of-the-art approaches [30][31].

Chapter 3

Camera Selection and Handoff as a Potential Game

3.1. Motivation and Problem Formulation

Game theory is well known for analyzing the interactions as well as conflicts among multiple agents [15][16]. Analogously, in a video sensor network, collaborations as well as competitions among cameras exist simultaneously. The cooperation lies in the fact that all the available cameras, those which can "see" the target person, have to collaborate to track the person so that the person can be followed as long as possible. On the other hand, the available cameras also compete with each other for the rights of tracking this person, so that a camera can maximize its own utility, as a camera's utility is closely related to how well it can track a person. This enlightens us to view the camera selection problem in a game theoretic manner. A game is the interactive process [17] among all the participants (players) of a game, who strive to maximize their utilities. The utility of a player refers to the welfare that the players can get in the game. In our problem, for each person to be tracked, there exists a multi-player game, with the available cameras being the players. If there are multiple persons in the system, this becomes a multiple of multi-player game being played simultaneously[26].

Vehicle-target assignment [13] is a classical multi-player game that aims to allocate a set of vehicles to a group of targets and achieves an optimal assignment.

Viewing the persons being tracked as “vehicles” while the cameras as “targets”, we can adopt the vehicle-target assignment model to choose the “best” camera for each person. In the following, we propose a game theory based approach that is well suited to the task at hand.

3.2. Game Theoretic Framework

Game theory involves utility, the amount of “welfare” an agent derives in a game. We are concerned with three utilities: 1) Global utility: the overall degree of satisfaction for tracking performance. 2) Camera utility: how well a camera is tracking persons assigned to it. 3) Person utility: how well a person is satisfied while being tracked by some camera. Our objective is to maximize the global utility while making sure that each person is tracked by the “best” camera. When competing with other available cameras, the cameras bargain with each other. Finally a decision is made for the camera selection based on a set of probabilities.

An overview of the approach is illustrated in Figure 3.1. Moving objects are detected in multiple video streams. Their properties, such as the size of the minimum bounding rectangle and other region properties (color, shape, location within FOV etc.) are computed. Various utilities are calculated based on the user-supplied criteria and bargaining processes among available cameras are executed based on the prediction of person utilities from the previous iteration step. The results obtained from the strategy execution are in turn used for updating the camera utilities and the person utilities until the strategies converge. Finally those cameras with the highest converged probabilities

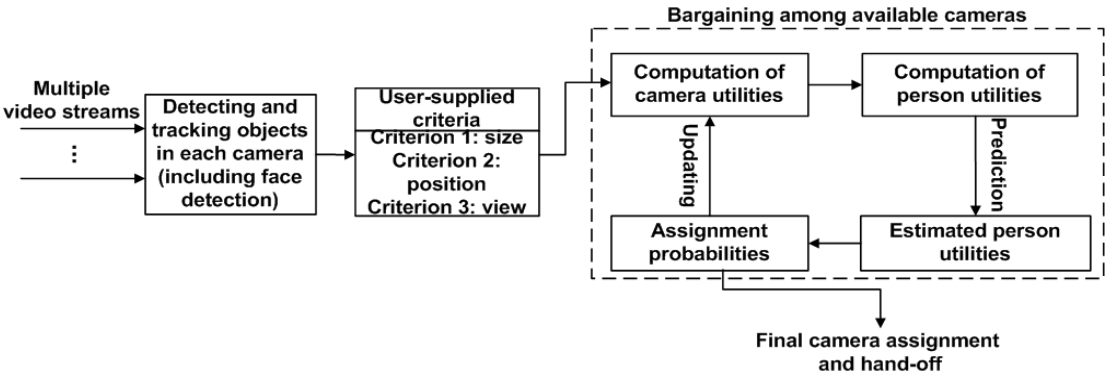


Figure 3.1: Game theoretic framework for camera selection and hand-off.

are used for tracking. This assignment of persons to the "best" cameras leads to the solution of the handoff problem in multiple video streams. A set of key symbols and their notations used in the following discussion are given in Table 3.1.

Table 3.1: Symbols and notations used in Chapter 3

Symbols	Notations
P_i	Person *i*
C_j	Camera *j*
N_P	Total number of persons in the entire network at a given time
N_C	Total number of cameras in the network at a given time
A_i	The set of cameras that can see P_i, $A_i = \{A_i^1, A_i^2, \dots, A_i^{n_C}\}$
n_C	Number of cameras that can *see* person *i*, number of elements in A_i
n_P	Number of persons currently assigned to camera C_j
a_i	The currently assigned "best" camera for person *i*
a_{-i}	The assignment of cameras for the persons excluding P_i
a	The assignment of cameras for all persons, $a = (a_i, a_{-i})$
$U_{C_j}(a)$	Camera utility for camera *j*
$U_{P_i}(a)$	Person utility for person *i*
$U_g(a)$	Global utility
$\bar{U}_{P_i}(k)$	Predicted person utility for person *i* at step *k*, $\bar{U}_{P_i}(k) = \left[\bar{U}_{P_i}^1(k), \dots, \bar{U}_{P_i}^l(k), \dots, \bar{U}_{P_i}^{n_C}(k)\right]^T$, where $\bar{U}_{P_i}^l(k)$ is the predicted person utility for P_i if camera a_l is used
$p_i(k)$	Probability of person *i*'s assignment at step *k*, $p_i(k) = \left[p_i^1(k), \dots, p_i^l(k), \dots, p_i^{n_C}(k)\right]^T$, where $p_i^l(k)$ is the probability for camera a_l to track person P_i

3.2.1. Computation of Utilities

We define the following properties of our system:

1. A person P_i can be in the FOV of multiple cameras. The available cameras for P_i belong to the set A_i. C_0 is a virtual camera that does not actually exist. We assume a virtual camera C_0 is assigned to P_i when there is no real camera in the network available to track P_i.
2. A person can only be assigned to one camera. The assigned camera for P_i is named as a_i.
3. Each camera can be used for tracking multiple persons.

We use a to denote the camera assignment for all the persons, and a_i denotes the assigned camera for P_i. For P_i, when we change the camera assignment from a_i' to a_i'' while assignments for other persons remain the same, if we have

$$U_{P_i}(a_i', a_{-i}) < U_{P_i}(a_i'', a_{-i}) \Leftrightarrow U_g(a_i', a_{-i}) < U_g(a_i'', a_{-i}) \quad (1)$$

the person utility U_{P_i} is said to be aligned with the global utility U_g, where a_{-i} stands for the assignments for persons other than P_i, i.e., $\mathrm{a}_{-\mathrm{i}} = (\mathrm{a}_1, \ldots, \mathrm{a}_{\mathrm{i}-1}, \mathrm{a}_{\mathrm{i}+1}, \ldots, \mathrm{a}_{\mathrm{N_P}})$. So, the camera assignment result a can also be expressed as $a = (a_i, a_{-i})$. We define the global utility as

$$U_g(a) = \sum_{C_j \in C} U_{C_j}(a) \quad (2)$$

where $U_{C_j}(a)$ is the camera utility and defined to be the utility generated by all the engagements of persons with a particular camera C_j.

Now, we define the person utility as

$$U_{P_i}(a) = U_g(a_i, a_{-i}) - U_g(C_0, a_{-i}) = U_{C_j}(a_i, a_{-i}) - U_{C_j}(C_0, a_{-i}) \quad (3)$$

where, C_0 is a virtual camera. The person utility $U_{P_i}(a)$ can be viewed as a marginal contribution of P_i to the global utility. To calculate (3), we have to construct a scheme to calculate the camera utility $U_{C_j}(a)$. We assume that there are N_{Crt} criteria to evaluate the quality of a camera used for tracking an object. Thus, the camera utility can be built as

$$U_{C_j}(a_i, a_{-i}) = \sum_{s=1}^{n_P} \sum_{l=1}^{N_{Crt}} Crt_{sl} \quad (4)$$

where n_P is the number of persons that are currently assigned to camera C_j for tracking and Crt are the criteria that are supplied by the user. Plugging (4) into (3) we can obtain

$$U_{P_i}(a_i, a_{-i}) = \sum_{l=1}^{N_{Crt}} (\sum_{s=1}^{n_P} Crt_{sl} - \sum_{\substack{s=1 \\ s \neq P_i}}^{n_P} Crt_{sl}) \quad (5)$$

where $s \neq P_i$ means that we exclude person P_i from the those who are being tracked by Camera C_j. One thing to be noticed here is that when designing the criteria, we have to normalize them. Besides this requirement, it does not matter what kind of criteria is used to be fed into the bargaining mechanism which is discussed below.

3.2.2. Criteria for Camera Selection and Handoff

The choice of a criterion to be used for camera selection and handoff depends on the users' requirements. There might be different criteria for different applications, such as criteria for power consumption, time delay, image resolution etc. The camera selection results may change due to applying different criteria. Our goal is to find the proper camera selection solution quickly based on whatever criteria are supplied by the user. In the following, we provide four criteria, which include human biometrics, which can be used for camera selection and handoff.

- Criterion 1: The size of the tracked person. It is measured by the ratio of the number of pixels inside the bounding box of the person to the size of the image. That is

$$r = \frac{\#\ of\ pixels\ inside\ the\ bounding\ box}{\#\ of\ pixels\ in\ the\ image\ plane}$$

Here, we assume that neither a too large nor a too small object is convenient for observation. Assume that λ is the threshold for best observation, i.e., when $r = \lambda$ this criterion reaches its optimal value.

$$Crt_{s1} = \begin{cases} \frac{1}{\lambda} r, & when\ r < \lambda \\ \frac{1-r}{1-\lambda}, & when\ r \geq \lambda \end{cases} \tag{6}$$

where $\lambda \in (0,1)$ is defined as the optimal ratio of the size of the minimum bounding box for the human body to the size of the image. These two sizes can be obtained by reading the coordinates of the bounding box and the size of the image. λ is dependent on the orientation of the camera and the location of a region-of-interest (ROI) in the image plane. Only in extreme rare situations a ROI will have a minimum width of one pixel. The value of λ remains valid at all times. Because of these reasons we do not do any camera calibration to find its extrinsic or intrinsic parameters. An example for the function Crt_{s1} when $\lambda = \frac{1}{15}$ is shown in Figure 3.2 as an illustration.

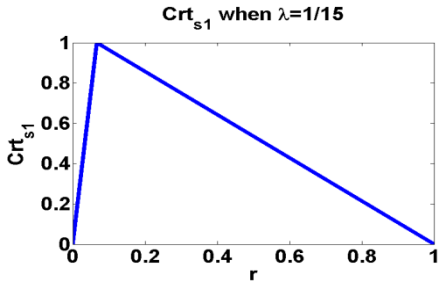


Figure 3.2: Function of Crt_{s1} when $\lambda = \frac{1}{15}$.

- Criterion 2: The position of a person in the FOV of a camera. It is measured by the Euclidean distance that a person is away from the center of the image plane

$$Crt_{s2} = \frac{\sqrt{(x-x_c)^2+(y-y_c)^2}}{\frac{1}{2}\sqrt{x_c{}^2+y_c{}^2}} \tag{7}$$

where *(x, y)* is the current position (body centroid) of the person and (x_c, y_c) is the center of the image.

- Criterion 3: The view of a person. It is measured by the ratio of the number of pixels on the detected face to that of the whole bounding box. That is

$$R = \frac{\#\ of\ pixels\ on\ the\ face}{\#\ of\ pixels\ on\ the\ entire\ body}$$

We assume that the threshold for the best frontal view is R , i.e., when $R = \xi$ ($\xi \in (0,1)$) the view of the person is the best.

$$Crt_{s3} = \begin{cases} \frac{1}{\xi} r, & when\ R < \xi \\ \frac{1-R}{1-\xi}, & when\ R \geq \xi \end{cases} \tag{8}$$

- Criteirion 4: Combination of criterion (1), (2) and (3). It is given by the following equation,

$$Crt_{s4} = \sum_{l=1}^{3} w_l Crt_{sl} \tag{9}$$

where w_l is the weight for different criterion.

It is to be noticed that all these criteria are appropriately normalized for calculating the corresponding camera utilities.

3.2.3. Bargaining Among Cameras

As stated previously, our goal is to optimize each person's utility as well as the global utility. Competition among cameras finally leads to the Nash equilibrium [16], as the

solution of the camera selection and handoff. Unfortunately, this Nash equilibrium may not be unique. Some of the solutions may not stable, which are not desired. To solve this problem, a bargaining mechanism among cameras is introduced, to make these cameras finally come to a compromise and generate a stable solution.

When bargaining, the assignment in the k^{th} step is made according to a set of probabilities

$$p_i(k) = \left[p_i^1(k), \dots, p_i^l(k), \dots, p_i^{n_C}(k)\right]^T$$

where $\mathrm{n_C}$ is the number of cameras that can "see" the person P_i and $\sum_{l=1}^{n_C} p_i^l(k) = 1$, with each $0 \le p_i^l(k) \le 1, l = 1, \dots, n_C$. We can generalize $p_i(k)$ to be

$$p_i(k) = \left[p_i^1(k), \dots, p_i^l(k), \dots, p_i^{N_C}(k)\right]^T$$

by assigning a zero probability for those cameras which cannot "see" the person P_i, meaning that those cameras will not be assigned according to their probability. Thus, we can construct an $N_C \times N_P$ probability matrix

$$\begin{bmatrix} p_1^1(k) & \dots & p_{N_P}^1(k) \\ \vdots & \ddots & \vdots \\ p_1^{N_C}(k) & \dots & p_{N_P}^{N_C}(k) \end{bmatrix}$$

At each bargaining step, we will assign a person to the camera which has the highest probability. We assume that one camera has no information of other cameras' utilities at the current step, which makes it hard to calculate all the possible current person utilities. So, we introduce the concept of predicted person utility $\overline{U}_{P_i}(k)$: Before we decide the final assignment profile, we predict the person utility using the previous person's utility information in the bargaining steps. As shown in (5), person utility

depends on the camera utility, so, we predict the person utility for every possible camera that may be assigned to track it. Each element in $\overline{U}_{P_i}(k)$ is calculated by (10)

$$\overline{U}_{P_i}^{l}(k+1) = \begin{cases} \overline{U}_{P_i}^{l}(k) + \frac{1}{p_i^l(k)}\left(U_{P_i}\big(a(k)\big) - \overline{U}_{P_i}^{l}(k)\right), a_i(k) = A_i^l \\ \overline{U}_{P_i}^{l}(k) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad , otherwise \end{cases} \tag{10}$$

with the initial state $\overline{U}_{P_i}^{l}(1)$ to be assigned arbitrarily as long as it is within the reasonable range for $\overline{U}_{P_i}(k)$, for $l = 1, \dots, n_C$. For the symbols used in Equation 10, note that A_i^l is the l^{th} camera that is in the set of available cameras for person P_i, which is different from C_l, the lth camera in the system. C_l can be in more than one available camera sets for different persons, while A_i^l is the l^{th} component in A_i, the set of available cameras for person P_i. It means that A_i^l is unique in the set for person P_i. Once these predicted person utilities are calculated, it can be proved that the equilibrium for the strategies lies in the probability distribution that maximizes its perturbed predicted utility [16],

$$p_i(k)'\overline{U}_{P_i}(k) + \tau H(p_i(k)) \tag{11}$$

where

$$H\big(p_i(k)\big) = -p_i(k)'log(p_i(k)) \tag{12}$$

is the entropy function and τ is a positive parameter belonging to [0,1] that controls the extent of randomization, where log means taking the log of every element of the column vector $p_i(k)$ and resulting in a column vector. The larger τ is, the faster the bargaining process converges; the smaller the τ is, the more accurate result we can get. So, there is a tradeoff when selecting the value of τ. We select τ, empirically, as 0.5 in our experiments. The solution of (11) is proved [16] to be

$$p_i^l(k) = \frac{e^{\frac{1}{\tau}\bar{U}_{P_i}^l(k)}}{e^{\frac{1}{\tau}\bar{U}_{P_i}^1(k)} + \cdots + e^{\frac{1}{\tau}\bar{U}_{P_i}^{n_C}(k)}} \quad (13)$$

After several steps of calculation, the result of $p_i(k)$ tends to converge. Thus, we finally get the stable solution, which is proved to be at least suboptimal [13].

3.2.4. Game Theoretic Algorithm

This overall algorithm is summarized in Algorithm 3.1.

The bargaining mechanism and the criteria are tightly integrated in the proposed game theoretic approach. The bargaining process is based on a set of criteria, since the utilities used to update in each bargaining step are calculated using these criteria. Note that different criteria imply different emphasis and the definition of error (see Section 3.3.3) depends on them.

Algorithm 3.1: Game theoretic camera selection and handoff

Input: Multiple video streams.

Output: A probability matrix for camera assignments are made.

Algorithm Description:

- At a given time, perform motion detection and get the selected properties for each person that is to be tracked.
- For each person and each camera, decide which cameras can "see" a given person P_i.
- For those which can "see" the person P_i, initialized the predicted person utility vector $\bar{U}_{P_i}$(l).

Repeat

1. Compute the Crt_{sl} for each available camera.
2. Compute the camera utilities $U_{C_j}(a)$ by (4).
3. Compute the person utilities $U_{P_i}(a)$ by (5).
4. Compute the predicted person utilities $\bar{U}_{P_i}(k)$ by (10).
5. Derive the strategy by $p_i(k)$ using (13).

Until The strategies for camera assignments converge.

- Do camera selection and handoff based on the converged strategies.

3.2.5. Discuss of Convergence

Define

$$M_P \coloneqq \max\{U_{P_i}(a): a \in A$$

$$m_P \coloneqq \min\{U_{P_i}(a): a \in A\}$$

$$\delta \coloneqq \min\{|U_{P_i}(a(1)) - U_{P_i}(a(2))|: a(1), a(2) \in A, a_{-i}(1)$$

$$= a_{-i}(2), |U_{P_i}(a(1)) - U_{P_i}(a(2))| > 0\}$$

$$N \coloneqq \min\{n \in \{1,2, \dots, N_P\}\}$$

To show the convergence of the bargaining mechanism, we have:

1. If $a(k)$ is not a NE, and
2. $a(k) = a(k+1) = \cdots = a(k+N-1)$
3. Let $a^* = (a_i^*, a_{-i}(k))$ be such that $U_g(a_i^*, a_{-i}(k)) > U_g(a_i, a_{-i}(k))$

   For some P_i and $C_j$and some $a^* \in A_i$. Then $U_g(k+N) > \frac{\delta}{2}$ will be proposed at step k+N with at least probability $(1-\varepsilon)^{N_P-1}$.

For a detail proof please refer to [13].

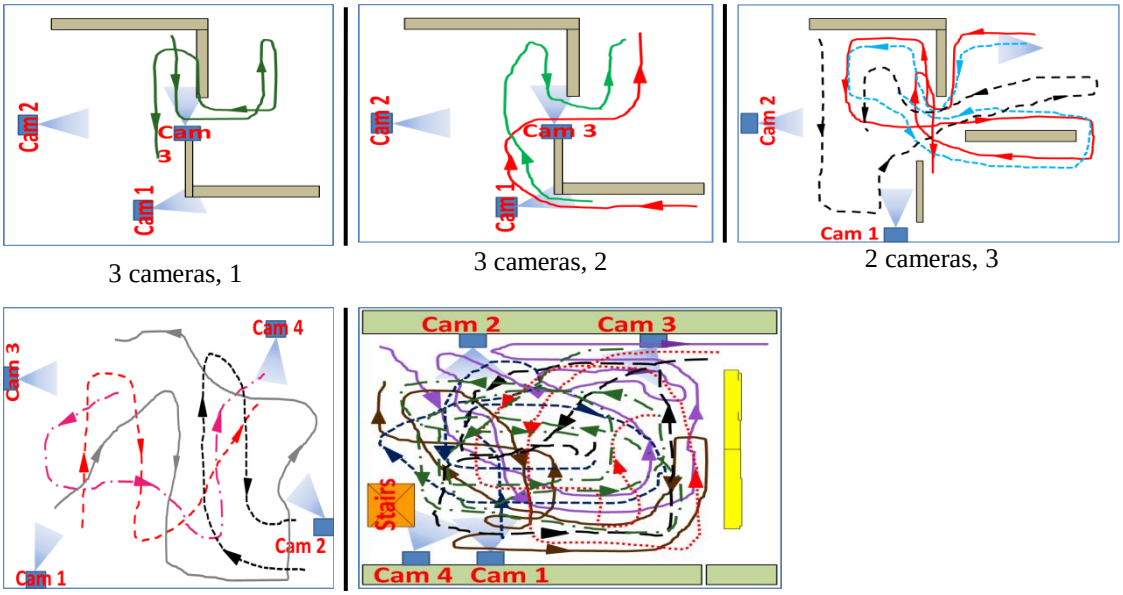
3.3. Experimental Results

3.3.1. Data and Parameters

A. Data

In our experiments, we tested the proposed approach on five cases: (1) 3 cameras, 1 person, (2) 3 cameras, 2 persons, (3) 2 cameras, 3 persons, (4) 4 cameras, 4 persons and (5) 4 cameras, 6 persons. These experiments include from the simple case, 3 cameras, 1 person, to a complicated case, 4 cameras, 6 persons. There are both cases with more

people than cameras (see Figure 3.14) and more cameras than people (see Figure 3.6 and Figure 3.10), which show that the performance of the proposed approach will not be influenced by relative numbers of cameras and persons. Both indoor and outdoor experiments are provided. The lengths of the video sequences vary from 450 frames to 700 frames. The frame rate for all indoor videos is 30 fps while that for outdoor videos is 15 fps. The cameras used in our experiments are all Axis 215 PTZ cameras, which are placed arbitrarily. To fully test whether the proposed approach can help to select the "best" camera based on the user supplied criteria, some of the FOVs of these cameras are allowed to interact while some of them are non-overlapping. The experiments are carried out in three different places with no camera calibration done before hand. The trajectories are randomly chosen by the persons for walking. We visualize the camera configuration and the persons' trajectories for the 5 cases in Figure 3.3.



3 cameras, 1 | 3 cameras, 2 | 2 cameras, 3

4 cameras, 4 | 4 cameras, 6

Figure 3.3: Camera configuration and the persons' trajectories in the experimented cases.

B. Parameters

In our experiments, we empirically give values to the parameters required by the criteria introduced in Section 3.2.2. $\lambda = \frac{1}{15}, \xi = \frac{1}{6}, w_1 = 0.2, w_2 = 0.1, w_3 = 0.7$.

3.3.2. Tracking and Face Detection

A. Tracking

All the experiments are conducted using the Continuous Adaptive Meanshift (Camshift) tracker [18] to evaluate the camera selection and handoff mechanisms. Theoretically, which tracker is used is not important as long as it can provide the tracking information that consists of size (size of the bounding box of a person) and location (position of the centroid of the bounding box) of a person. It is to be noticed that the same tracker is used for all the experiments and all the camera assignment approaches that are compared to filter out the influence of a tracker to the camera selection results.

The walking persons are initially selected by an observer manually when a person enters the FOV of a camera as detected by the background subtraction method. The persons who participated in the experiments wear clothes in distinct colors, so different persons can be identified by calculating the correlation of the hue histograms of the pixels inside their bounding boxes (ROIs) using the function CompareHist [19].

- Errors Caused by the Tracker ($N_c = 2, N_p = 3$, Indoor)

There are some errors that are caused by the failure of the tracker. In Figure 3.4, we show some error frames in a 2 cameras, 3 persons case, which are due to the failure of the Camshift tracker. The Camshift tracker is not robust when severe occlusion happens

and it can be distracted by the object with similar colors as the target. However, the camera assignment results are correct if we ignore the errors that are caused by the tracker, i.e., if we assume that the tracker provides a correct ROI for the target, then the camera assignments, performed based on the user-supplied criteria, are correct. For instance, in Figure 3.4 (4-1 and 4-2), the system should select camera 1 to track the person in red, where the person has a frontal view which is preferred according to the user-supplied criteria. However, the tracker for the person in red is distracted by the red pillow, which causes error for the camera assignment. But if we assume that the ROIs returned by the tracker are correct, then based on the size and position of the person in red (frontal face is not available because of the wrong tracking result), the system selects the correct camera.

Figure 3.4: An example for the failure of the Continuous Adaptive Meanshift (Camshift) tracker. We only draw ROIs in the cameras that are selected to track the persons. We generate bounding boxes only for those cameras that are selected to track a person. In 4-1, camera 1 is distracted to the red pillow, such that the system selects camera 2 for the person in red, which is an error based on the error definition in Section 3.3.3.

Camera 1	Camera 2

B. Face Detection

Face detection is done in a particular region (top 1/3 height of the ROI), provided by the tracker. We use the cascade of haar feature based classifiers for face detection [19]. It can detect faces correctly in 90%+ cases when the tracker returns a correct ROI.

3.3.3. Performance Measures

In our experiments, the bottom line is to track walking persons seamlessly, i.e., the system will follow a person as long as the person appears in the FOV of at least one camera. In the case where more than one camera can "see" the persons, we assume that the camera that can "see" the person's face is preferable. This is because in surveillance systems, the frontal view of a person can provide us more interesting information than other views. So, based on this criterion, we define the camera assignment error in our experiments as: (1) failing to track a person, i.e., a person can be seen in some cameras in the system but there is no camera assigned to track the person, or (2) failing to get the frontal-view of a person whenever it is available. We define these error terms in the following:

N_{lost} - the number of times that a target person is lost. It is determined if the bounding box returned by the tracker covers less than 30% of the person's actual size or is larger than 150% of the person's actual size during tracking. The term region-of-interest (ROI) and bounding box are used interchangeably in this chapter.

N_{fvl} - the number of times a frontal view is detected but not selected by any camera. Note that in our experiments, there is no case where a frontal view is detected but the person is lost during tracking. So, the intersection of the above two cases should be

empty, i.e. $N_{lost} \cap N_{fvl} = \emptyset$.

$NP_i^{C_j}$- the number of persons appearing in Camera C_j in frame i.

N_f – the total number of frames in an experimented video.

NPC_i - the number of cameras with no persons in frame i.

N_C - the total number of cameras in an experiment.

The total error of a video is defined as

$$Err = N_{lost} + N_{fvl} \tag{14}$$

The error rate is defined as the error normalized by total the numbers of cameras and persons in all frames. Frames in which there are no persons are counted as correct frames, since there are no errors caused by losing a person or lose the frontal view of a person. Frames with more than one person in the FOV of a camera are multiply counted to normalize by the multiple persons. Error rate ER is defined as

$$ER = \frac{Err}{\sum_{i=1}^{N_f}\sum_{j=1}^{N_C} NP_i^{C_j} + \sum_{\substack{i=1 \\ NP_i^{C_j}=0}}^{N_f} NPC_i} \tag{15}$$

3.3.4. Evaluation of Game Theoretic Framework

A. Experiment #1: Criterion Selection ($N_C = 3, N_P = 2$, Indoor)

Since there are multiple criteria to be used in the experiments, we first test the performance for different criterion in a 3 cameras, 2 persons case. A general description of the videos is shown in Table 3.2.

Different experiments are carried out using the single and the combined criterion described in Section 3.2.2. Some typical results are shown in Figure 3.5. To make it

Table 3.2: Experiment #1. Overview of videos for each camera and the number of handoffs that are taken place (Nof: number of frames, results are shown as mean(standard deviation)

	Nof (0 person)	Nof (1 person)	Nof (2 persons)	Nof (with occlusion)	No. of handoffs (Crt_{s4})
Cam1	56	22	12	0	2 (0)
Cam2	14	46	18	11	9(1)
Cam3	44	23	17	6	6(1)

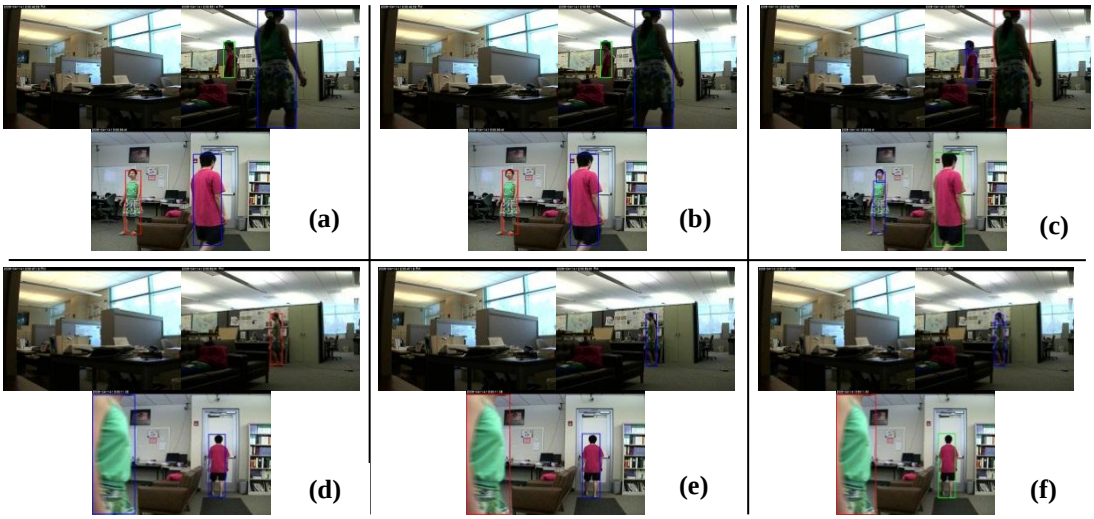


Figure 3.5: Experiment #1. A comparison for using different criteria. The first row and the second row are for two time instants respectively. The first column through the third column are using criterion 1 to criterion 3, respectively.

Camera 1 | Camera 2

Camera 3

convenient for a comparison, we show the tracking results for other cameras as well, no matter whether they are selected for tracking or not. The cameras, for which the bounding boxes are drawn in blue, are selected for tracking while the ones in red or green are not as good as the blue ones.

Figure 3.5 (a) to (c) use criterion 1 to 3 at time instant 1 while (d) to (f) use criterion 1 to 3 at time instant 2. It can be observed from Figure 3.5 (d) that the problem for using criterion 1 only is that when the persons are getting close to the cameras, the size of the bounding box increases, and while the resolution is not very high, persons are

not clear enough. Meanwhile, there are cases such that when a person is entering the FOV of a camera, the size of the person is not small but only part of the body is visible. This should not be preferred if other cameras can give a better view of the body. Thus, we introduced criterion 2, considering the relative position of persons in the FOVs of the cameras. The closer the centroid of a person is to the center of the FOV of a camera, the higher the camera utility is generated. We can observe that when applying criterion 2 in Figure 3.5 (e), the camera with the person near the center is chosen and we can obtain a higher resolution of the person compared to the results based on criterion 1 in Figure 3.5 (d). However, the problem for using criterion 1 or criterion 2 only is that we reject the camera(s) which can see a person's face, which is of general interest. This case is shown in Figure 3.5 (a) (b) and (d). To solve this problem, we developed criterion 3 (the view of the person). So, when applying criterion 3, we obtain a more desirable camera with a frontal view of the person in Figure 3.5 (c) and (f). Whereas criterion 3 can successfully select a camera with a frontal-view person, it may fail to track a person when no face can be detected. As shown in Figure 3.5 (f), although the person is in the FOV of some camera, the person is lost based on criterion 3.

So, finally, we come up with a weighted combination of these three criteria. As stated previously, we use 0.2, 0.1 and 0.7 as the weights for these three criteria respectively so that, in most cases, the system will choose the camera which can "see" a person's face. For those frames where there is person without the detected face, the combination criterion can also provide the "best" camera based on criteria 1 and 2 and, thus, realizing continuous tracking. All the camera handoffs, when applying the

Figure 3.6: Experiment #1. All camera hand-offs when applying the combined criterion for 3 cameras, 2 persons case. The cameras that are selected for tracking a person provides a blue bounding box for that person, otherwise it provides green bounding box for the person in red and red bounding box for the person in green.

Camera 1	Camera 2
Camera 3	

combined criterion, are shown in Figure 3.6. The error rate in this case is 5.56%, while that for using criterion 1 to 3 only are 25.56%, 10.00% and 30.00%, respectively.

The number of handoffs in this 3 cameras, 2 persons case is give in Table 3.2. Camera utilities, person utilities and the corresponding assignment probabilities for the using the combined criterion is shown in Figure 3.7, where Probability[i][j] stands for the probability that C_j is assigned to track P_i.

We use the combined criterion for all the other experiments in the rest of this chapter.

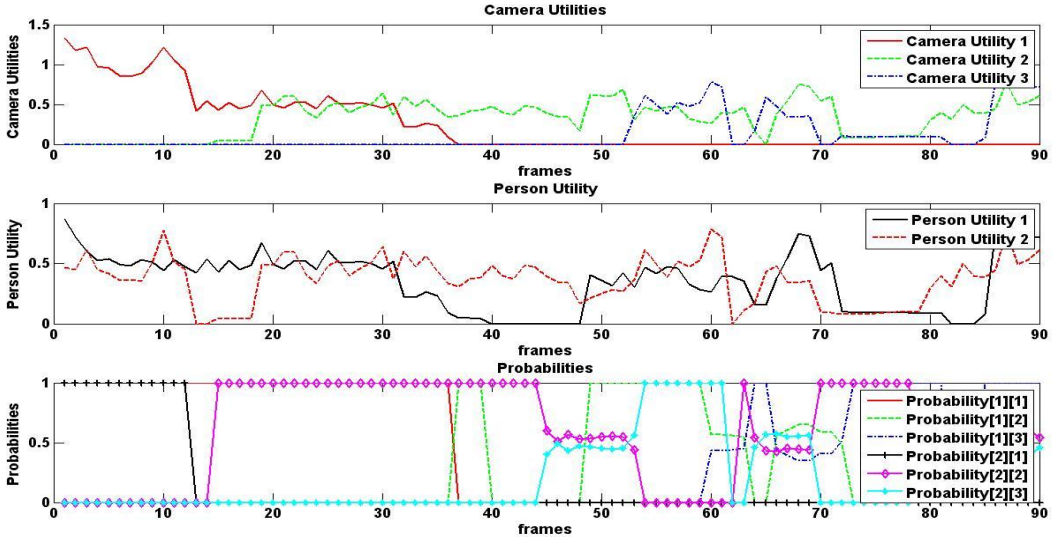


Figure 3.7: Experiment #1. Utilities and assignment probabilities for each processed frame when using the combined criterion.

B. Convergence of Results for Bargaining

For the above experiments, in most cases, the probabilities for making the assignment profile converges (with $\varepsilon < 0.05$, where ε is the difference between the two successive results) within 5 iteration. So we use 5 as the number of iterations threshold when bargaining. Thus, for those cases that will not converge within 5 iterations, there may be an assignment error based on the unconverged probabilities. In Figure 3.8 we plot the number of iteration with respect to every processed frame for Experiment #1. It turns out that the average number of iterations is 1.37. As the numbers of persons and cameras increase, this bargaining system will save a lot of computational cost to get the optimal camera assignments. A typical convergence for one of the assignment probabilities in a bargaining among cameras is given in Figure 3.9. We also show an example of error caused by the failure of the bargaining mechanism in a more complicated (4 cameras, 6 persons) Experiment #6 discussed later in the comparison part.

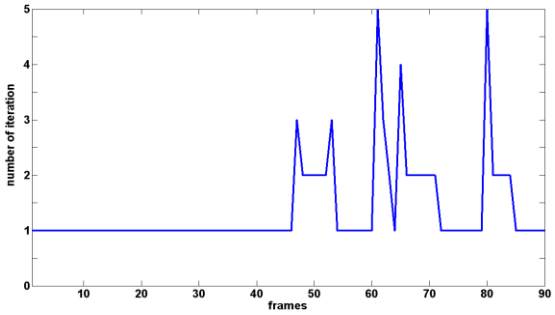


Figure 3.8: Experiment #1. Number of iteration for the bargaining mechanism in each frame.

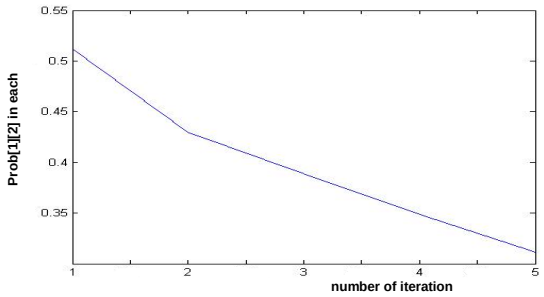


Figure 3.9: Experiment #1. A typical convergence in the bargaining process (Frame 56, camera 2, for the person in green).

3.3.5. Comparison of Game Theoretic Approach with Other Related Approaches

In this section, we will compare our approach with two other approaches: the first approach [3] performs camera handoff by calculating the co-occurrence to occurrence ratio (COR). We will call this the COR approach. The second approach performs the camera selection problem by solving the Constraint Satisfaction Problem (CSP) [4]. We will call this approach the CSP approach in the following. As concluded in Section 3.3.4, we will use the combined criterion (Equation 9) for the following comparisons.

A. Comparison with the COR Approach

In [3], the mean probability that a moving object is detected at a location x in the FOV of a camera is called an occurrence at x. The mean probability that moving objects are simultaneously detected at x in the FOV of one camera and x' in the FOV of another camera is called a co-occurrence of x and x'. The COR approach decides whether two points are in correspondence with each other by calculating the co-occurrence to occurrence ratio. If the COR is higher than some predefined threshold, then the two points are decided to be in correspondence with each other. When one point is getting

close to the edge of the FOV of one camera, the system will handoff to another camera that has its corresponding point. However, the COR approach in [3] has been applied to two cameras only. We generalize this approach to the cases with more cameras by comparing the accumulated COR in the FOVs of multiple cameras. We randomly select 100 points on the detected person, train the system for 10 frames to construct the correspondence for these 100 points, calculate the cumulative CORs in the FOVs of different cameras and select the one with the highest value for handoff.

Experiments have been done to compare the COR approach with our approach for the 3 cameras, 1 person case (Experiment #2) and the 3 cameras, 2 persons case (Experiment #3).

- Experiment #2: Comparison with COR Approach ($N_C = 3, N_P = 1$, Indoor)

The handoff process by using the COR approach and the corresponding frames by using our approach (may not be the handoff frames) are shown in Figure 3.10. In Figure 3.10 (g) to (h), the COR approach switches to camera 1, while our proposed approach sticks to camera 2 (Figure 3.10 (c) to (d)) to get the frontal view of the person. The COR approach needs a time period to construct the correspondence between different views. We let this period to be 10 frames. As a result, there is some time delay for the handoff. For instance, in Figure 3.10 (a) to (b), our approach has already selected camera 3 in (a), where a frontal view of the person is already available and the size of the person is acceptable, while the COR approach switched to camera 3 in (d) when the person is detected as leaving the FOV of camera 2 and entering the FOV of camera 3.

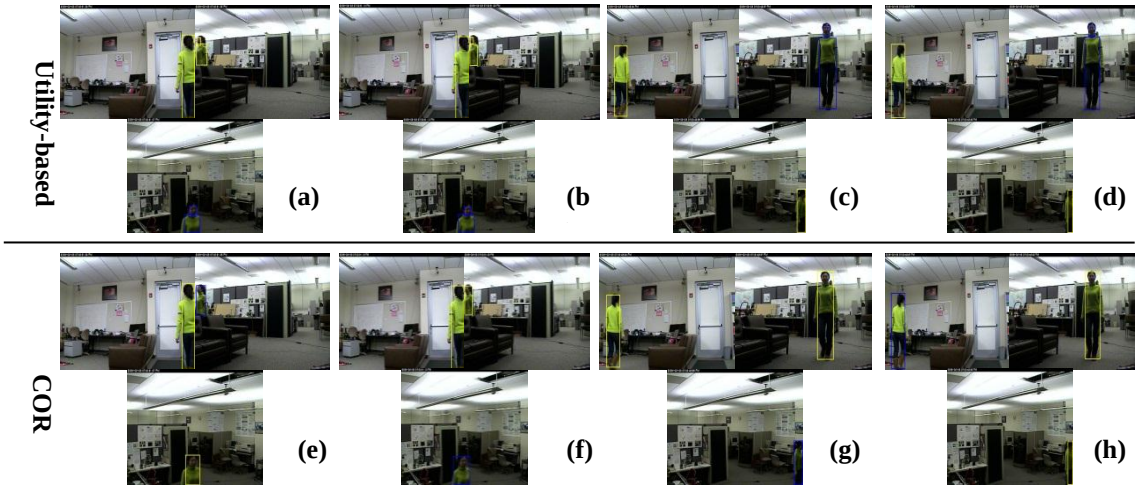


Figure 3.10: Experiment #2. Two camera hand-offs by using the co-occurrence to occurrence ratio (COR) approach and the comparison with our approach. The first row are the results by our approach and the second row are the results by the COR approach. The camera selected to track the person provides a blue bounding box, otherwise it provides a yellow bounding box.

Camera 1	Camera 2
Camera 3	

• Experiment #3: Comparison with the COR Approach ($N_C = 3, N_P = 2$, Indoor)

In Figure 3.11, we show some error frames by using the COR approach. These results can be compared with Figure 3.6 (Experiment #1) where we use the same video for the proposed approach. By the comparison, we can notice that the COR approach can only switch the camera to another one when the person is about to leave the FOV, but cannot select the "best" camera based on other criteria. So, the number of handoffs by our approach is larger than that of the COR approach (See Table 3.3). If we use the definition of error as stated in Section 3.3.3, the error rates for these two cases are compared in Table 3.3. Based on this error definition, the COR approach loses the frontal view of a person more easily. Examples are Figure 3.11 (b) (lose the person in red), (d) (lose the frontal view of the person in red) and (f) (lose the frontal view of the person in green).

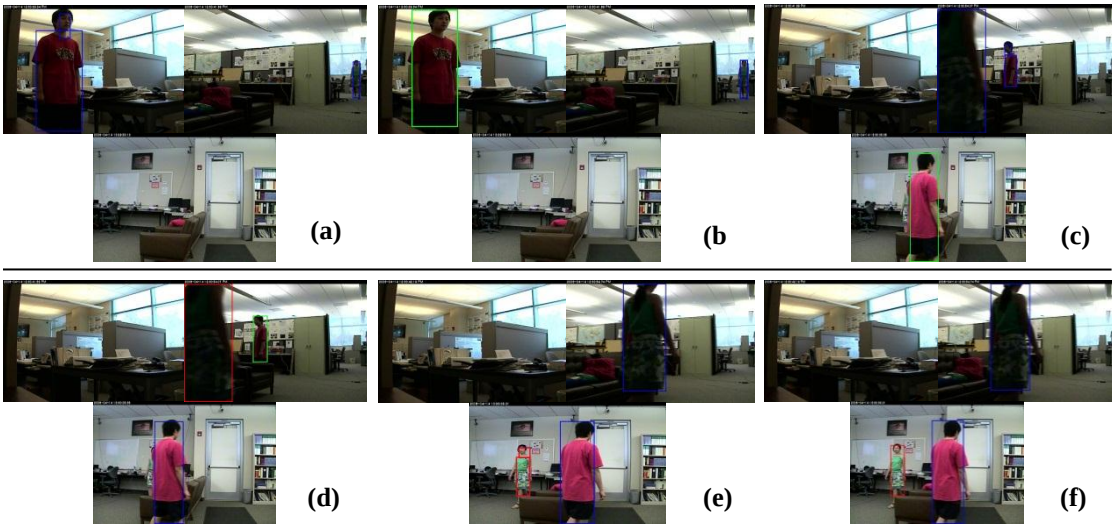


Figure 3.11: Experiment #3. Some camera hand-off errors by the co-occurrence to occurrence ratio (COR) approach in a 3cameras, 2 persons case. The cameras that are selected for tracking a person provides a blue bounding box for that person, otherwise it provides green bounding box for the person in red and red bounding box for the person in green.

Table 3.3: Comparison of error rates for the co-occurrence to occurrence ratio (COR) approach and the proposed approach. Results are shown as mean (standard deviation)

	Experiment #2		Experiment #3	
	# of handoffs	Error rate	# of handoffs	Error rate
COR	4(1)	38.77% (2.32%)	5(0)	45.62% (0)
Proposed	6(0)	2.87%(0.05%)	8(1)	5.56% (0.12%)

B. Comparison with the CSP Approach

The approach in [4] solves the camera selection problem by using the constraint satisfaction approach. According to the key assumptions made in Section 3.2.1, we allow one camera to track multiple persons but one person can only be tracked by one camera. So, for each camera C_j, we let all those persons that can be seen by this camera form a group g_j. For instance, if, in our case, the camera C_j can see person P_1 and P_2, then the

domain of g_j , noted as $Dom[g_j]$, is $\{\{P_1\}, \{P_2\}, \{P_1, P_2\}\}$. The constraint is set to be $d_i \cap d_j = \{\emptyset\}$, for $i \neq j$, where $d_i \in b_i \cup \emptyset$ is the camera assigned to track person P_i , and b_i and b_j belong to $Dom[g_j]$ and $i \neq j$. By doing so, we mean that the persons to be tracked are assigned to different cameras. We changed some of the notations from [4] so that the notations in this section are not in conflict with the notations used in the previous sections of this chapter.

Experiments for 3 cameras, 2 persons (Experiment #4) and 4 cameras, 4 persons (Experiment #5) cases are carried out under the above constraint to maximize the criterion 4 (Equation 9), using the BestSlov algorithm in [4].

- Experiment #4: Comparison with the CSP Approach ($N_C = 3, N_P = 2$, Indoor)

Since our approach requires 5 iterations for the 3 cameras, 2 persons case (Experiment #3) to get acceptable results, we also use 5 backtracking steps in the CSP approach.

Both the CSP approach and our proposed approach are able to accommodate different criteria. Most of the time, the CSP approach can select the "best" camera, based on our criterion and the error definition. So, we only compare the number of handoffs and the error rates for this case in Table 3.4. The results show that the CSP approach has higher error rates than our approach.

Table 3.4: Comparison of error rates for the constraint satisfaction problem (CSP) approach and the proposed approach. Results are shown as mean (standard deviation)

	Experiment #4		**Experiment #5**	
	# of handoffs	Error rate	# of handoffs	Error rate
CSP	9 (1)	8.38% (1.26%)	17 (3)	10.66% (3.78%)
Proposed	8 (1)	5.56% (1.26%)	19 (2)	7.32% (2.51%)

• Experiment #5: Comparison with the CSP Approach ($N_C = 4, N_P = 4$, Indoor)

Since in this case, there are more persons and cameras involved, we increase the number of backtracking steps and the number of iterations to 10. Because the performance of the CSP approach heavily depends on the number of backtracks (the more backtracks it takes, the more accurate the results can be), as the number of cameras and persons goes up, the CSP approach will miss the "best" camera with a high probability. Some of the errors for this case are shown in Figure 3.12. There are errors when a person's frontal view is available but it is not chosen such as in Figure 3.12 (b) (d) and (f), or when a person's frontal view is unavailable, the system chooses the camera with a smaller size person and farther from the center of the FOV, such as in Figure

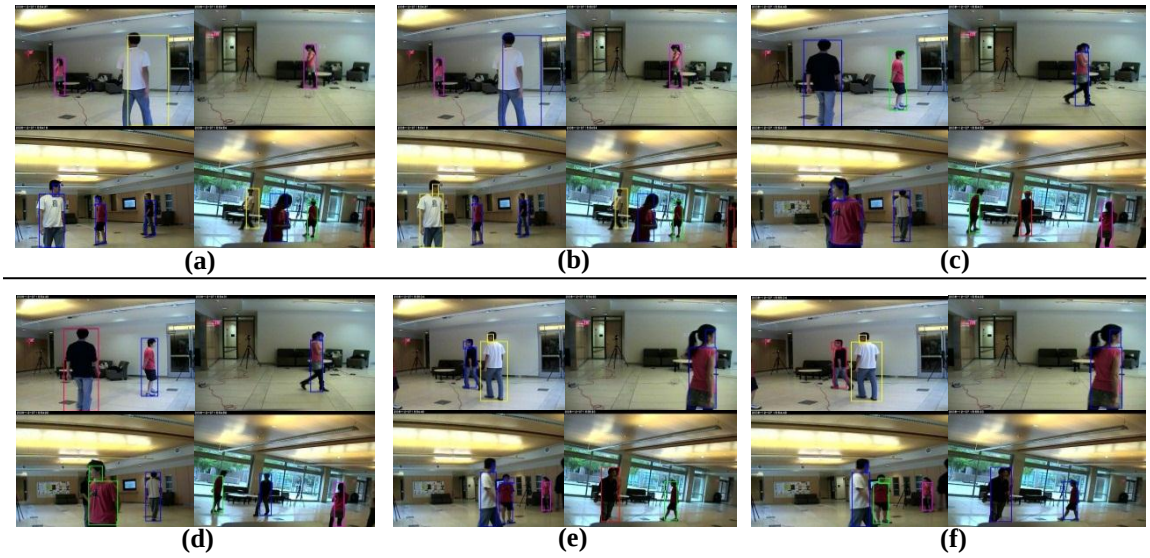


Figure 3.12: Experiment #5. Some error frames by using the Constraint Satisfaction Problem (CSP) approach for the 4 cameras, 4 persons case. The camera with blue bounding box for a person is selected to track the person. The cameras selected for tracking a person provides a blue bounding box for that person.

Camera 1	**Camera 2**
Camera 3	**Camera 4**

3.12(d) for the person in black. The high error rate for the CSP approach is due to its computational cost.

• Experiment #6: Further Comparison between the CSP and the Proposed Game Theoretic Approach – Number of Iterations ($N_c = 3, N_P = 1 - 10$)

Figure 3.13 gives a comparison of number of iterations for our approach and the number of backtracks for the CSP approach for the case when the number of cameras is fixed to 3 and the number of persons goes up from 1 to 10. We can see that although the CSP approach can solve the camera selection problem based on different user-defined tasks, it is computationally expensive as the complexity of the system increases.

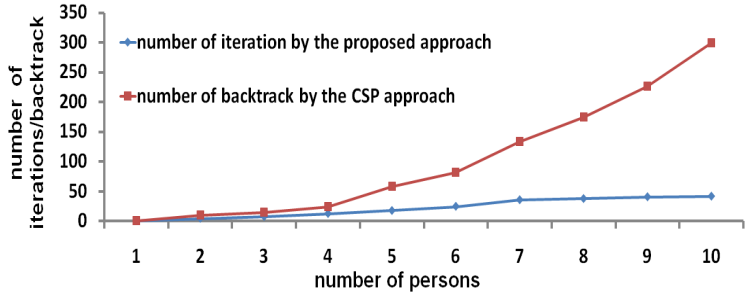


Figure 3.13: Comparison for the number of iteration or backtrack by the proposed utility-based approach and by the Constraint Satisfaction Problem (CSP) approach.

C. Comparison among Game Theoretic, COR and CSP Approaches

• Experiment #7: $N_C = 6, N_P = 4,$ Outdoor

In this section, we consider a more complicated case with 4 cameras and 6 persons. Because there are too many people in the system, it will be hard to observe if we mark the person in all the cameras that can see them. So, we only draw the bounding boxes for those cameras which are assigned to track the specific person. Different colors

are used to distinguish different persons. We only display some typical results (Figure 3.14) for each of the approaches that are compared. Because there are more cameras and persons involved in this experiment than the previous ones, we increase the number of iterations to 20 for all the CSP and the proposed approach.

For the proposed approach, we can notice that whenever there is a camera available to track a specific person, the camera selection can be performed based on the pre-defined criteria. In Figure 3.14 A6 (the Utility-based approach group), we provide a case when the bargaining mechanism fails, i.e., the number of iterations is not large enough to converge to the optimal result. In this figure, the person in red bounding box should be tracked by Camera 1 based on an exhaustive calculation which can be regarded as the ground-truth.

The COR approach cannot decide which camera to select based on the user supplied criteria. So most of the handoffs take place when a person is leaving the FOV of one camera and entering the FOV of another camera. The CSP approach can deal with the supplied criteria to some extent, but since 20 backtracks are too few to reach the optimal answer, the CSP loses the "best" camera easily.

The overall performance of these approaches is presented in Table 3.5.

Table 3.5 Comparison of error rates for the COR, CSP and the proposed approach. Results are shown as mean (standard deviation)

Experiment #6	COR	CSP	Proposed
Error rates	45.67% (2.56%)	12.96% (1.67%)	7.89% (0.98%)

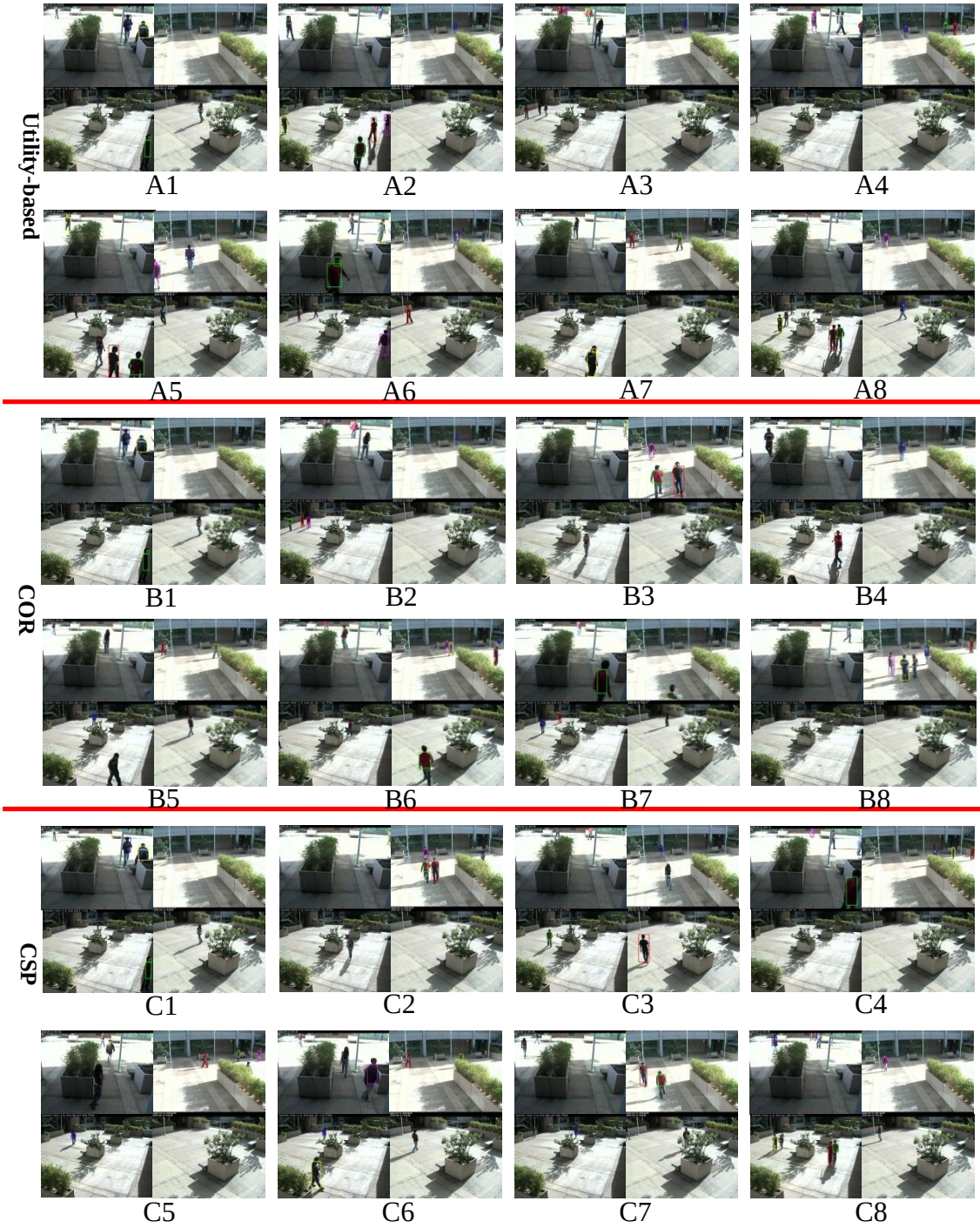


Figure 3.14. Experiment #6. A comparison for the proposed utility-based game theoretic approach, the COR approach and the CSP approach. Only those cameras selected to track a person provide a bounding box for that person.

Camera 1	**Camera 2**
Camera 2	**Camera 4**

3.4. Summary

In this chapter, we proposed a new principled approach based on game theory for the camera selection and handoff problem. We developed a set of intuitive criteria in this chapter and compared them with each other as well as the combination of them. Our experiments showed that the combined criterion is the best based on the error definition provided in Section 3.3.3. Since the utilities, input of the bargaining process, largely depend on the user-supplied criteria, our proposed approach can be task-oriented. Unlike the conventional approaches which perform camera handoffs only when an object is leaving or entering the FOV, we can select the "best" camera based on the pre-defined criteria.

The key merit of the proposed approach is that we use a theoretically sound game theory framework with bargaining mechanism for camera selection in a video network so that we can obtain a stable solution with a reasonably small number of iterations. The approach is independent of (a) the spatial and geometrical relationships among the cameras, and (b) the trajectories of the objects in the system. It is robust with respect to multiple user-supplied criteria. The approach is flexible since there is no requirement for a specific criterion that a user is obligated to use. A wide variety of experiments show that our approach is computationally more efficient and robust with respect to other existing approaches [3][12].

We analyzed the influence of a tracker on the proposed approach in Section 3.3.2 and compared our work with two other recent approaches both qualitatively and quantitatively. All the experiments used a physical camera network with real data in real

time. This included both indoor and outdoor environments with different numbers of cameras and persons. As compared to the other approaches, it is shown that the proposed approach has smaller error rates in all the experiments. The computational efficiency of the proposed approach is also verified quantitatively. This comparison shows that (a) COR approach cannot do any criterion-dependent camera selection. (b) As the number of cameras and persons in the system increases, the selection ambiguity and failure also increase in the COR approach. (c) The CSP approach is task-dependent and can select the "best" camera based on whatever criterion is provided by the user. (d) The CSP approach is computationally much more expensive than our approach.

Chapter 4

Camera Selection and Handoff as a Weakly Acyclic Game

4.1. Assumptions, Symbols and Notations

To keep the completeness and independency of each chapter, we may provide assumptions, symbols and notations in each chapter, which is independent of those used in other chapters.

4.1.1. Assumptions

We make the following assumptions for the proposed system for camera selection and handoff in this chapter:

1) One camera can be assigned to track multiple persons. However, one person can be only tracked by the "best" camera (see Section 4.3.3) of the person. We make this assumption to free up more cameras so that we can have a larger coverage or more objects can be monitored.

2) The communication among cameras is allowed and assumed to be noise-free. This allows for distributed computations. For camera selection and handoff, distributed system is more bandwidth efficient than a centralized system since only limited information is transmitted. Note that if there is no communication among cameras, then all the information has to be sent to a central server. In this case, a final decision is made by the central sever and sent back to each of the camera. However, this is not necessary if

communication among cameras is allowed. In our system, one camera's strategy is based only on its payoff, which may only be affected by other cameras' actions when a selected person (object) is visible in the FOV of multiple cameras. The observations of the cameras' payoffs are based on the image information that we can get from the video sequences, such as the size of the bounding boxes etc. So, in our system, we allow all the cameras to broadcast their status.

3) The trajectories of persons are not known. The objects in FOVs of different cameras are put in correspondence by comparing their feature vector (color and texture) correlations and the pre-calculated homographies.

4) A rough map of topology of the cameras is known. Homographies between overlapping view cameras are pre-calculated. The word "rough" means that we do not require detail position of each camera, but only which group it belongs to. Group of cameras are useful to accommodate different needs in different area of a video network. No full camera calibration is needed [55].

5) For each camera, the possible actions are sleep (a camera is set to sleep mode when no objects are expected in its FOV), awake – free of task (a camera is awakened up once its neighboring border camera(s) informs it that there is an object it may see, but it will not necessarily be assigned to follow that object) and awake – recording (a camera is assigned to follow an object and set to record mode to process and store the video). This can be viewed as strategies a player can take in a game.

4.1.2. Symbols and Notations

In order to describe our ideas more conveniently, we first provide the symbols and

Table 4.1: Symbols and notations used in Chapter 4

SYMBOLS	NOTATIONS	SYMBOLS	NOTATIONS
P_i	Person *i*	T_i	Value calculated for Equation (6)
C_j	Camera *j*	$\boldsymbol{a}^b$(t)	The cameras' baseline action vector at time *t*
N_P	Total number of persons	$\boldsymbol{p}^b(t)$	The baseline camera payoff vector at time *t*
N_C	Total number of cameras	$[\boldsymbol{a}^*, \boldsymbol{p}^*]$	A Nash Equilibrium in our problem
$\boldsymbol{A}_j$	Action set of camera *j*	$PO_{C_j}(\boldsymbol{a}(t))$	Camera *j*'s payoff when the camera assignment is $\boldsymbol{a}(t)$
n_C	Number of cameras that can *see* person *i*	$PO_g(\boldsymbol{a}(t))$	Global payoff when the camera assignment is $\boldsymbol{a}(t)$
n_P	Number of persons currently assigned to camera *j*	$PO_{P_i}(\boldsymbol{a}(t))$	Person *i*'s payoff when the camera assignment is $\boldsymbol{a}(t)$
$a_j(t)$	The action of camera *j* at time *t*	$PO_{C_j}(\boldsymbol{a}(t))$	Payoff function for camera *j* at time *t* for action $\boldsymbol{a}(t)$
$\boldsymbol{a}_{-j}(t)$	The action of all cameras other than camera *j* at time *t*	$PO_{C_j}^{\ b}(t)$	Baseline payoff of camera *j* at time *t*
$\boldsymbol{a}(t)$	Strategy, assignments of actions for all cameras at time *t*. $\boldsymbol{a}(t) = \{a_j(t), \boldsymbol{a}_{-j}(t)\}$	ε	Exploration rate, whose range is [0,1]
$U_j(\boldsymbol{a}(t))$	Player (camera in our case) *j*'s objective function with strategy $\boldsymbol{a}(t)$	δ	Improvement step, whose range is [0, 1]
$\phi(\boldsymbol{a}(t))$	Potential function with strategy $\boldsymbol{a}(t)$	(x, y)	Coordinate of the object's position
$a_j^b(t)$	Baseline (b) action for camera *j* at time *t*	(x_c, y_c)	Coordinate of the image plane center
S_{im}	The m^{th}criterion satisfaction for person P_i	N_{iter}	Number of iterations
S_{imC}	S_{im} in the C^{th} frame	S_{lost}	Error terms. See Section 4.4.2.
w_m	Weight for criterion S_{im}	S_d	Error terms. See Section 4.4.2.
w_{mC}	w_m in the C^{th}frame	S_{fl}	Error terms. See Section 4.4.2.
λ	Threshold for best size of the person	S_{gfl}	Error terms. See Section 4.4.2.
γ	Threshold for best view of the person	$Num\{S\}$	Error terms. See Section 4.4.2.
C	Current frame number used in S_{i5}	Err_i	Error in Application *i*
K	Last frame number used in S_{i5}	ER_i	Error rate in Application *i*
D	Threshold calculated for S_{i5}	N_F	Number of frames in a video

Table 4.2: Definitions for related terminologies

Terminologies	Definition
Game	A game can be any situation that involves two or more agents. In this paper, we view the process of selecting a "best" camera for a specific person as a game.
Player	All the agents in a game. In this paper, the players are the cameras that can see the person for whom the game is being played. When there are multiple persons, this will be a multiple of multi-player game.
Rational player	A player is rational means that the player makes decisions consistently by maximizing the player's own welfare.
Strategy	Camera C_j's strategy is the action $a_j(t)$ that it is going to take.
Utility	The welfare a player can get from the game.
Payoff	Payoff has the same meaning as utility. They both rank the desirability of a player to play a strategy.
Camera payoff $PO_{C_j}(\boldsymbol{a}(t))$	Used to rank a camera's desirability for its possible actions. $PO_{C_j}(\boldsymbol{a}(t))$ is Camera C_j's payoff when the camera assignment is $\boldsymbol{a}(t)$.
Local payoff	Local payoff is the camera payoff, $PO_{C_j}(\boldsymbol{a}(t))$.
Person payoff $PO_{P_i}(\boldsymbol{a}(t))$	Used to rank a person's desirability of being tracked by a camera. $PO_{P_i}(\boldsymbol{a}(t))$ is Person P_i's payoff when the camera assignment is $\boldsymbol{a}(t)$.
Global payoff $PO_g(\boldsymbol{a}(t))$	Evaluate the overall performance of the system. $PO_g(\boldsymbol{a}(t))$ is the global payoff when the camera assignment is $\boldsymbol{a}(t)$.
Nash Equilibrium	A set of strategies, one for each player, such that no player has incentive to unilaterally change its action.
Potential function	The global utility function.
Potential game	A game is a potential game if all players' payoff functions are aligned with the potential function, i.e., if a player changes its strategy, the increment of its payoff function is the same as that of the global payoff function. This is true for every player in a potential game.
Weakly acyclic game	A game is weakly acyclic if for a set of strategies, which is not a Nash Equilibrium, there is a better-reply path that leads to a Nash Equilibrium
Better-reply path	A better-reply path is a sequence of camera actions, $\boldsymbol{a}(1), \boldsymbol{a}(2), \dots, \boldsymbol{a}(T)$, such that for each successive steps $\boldsymbol{a}(t), \boldsymbol{a}(t+1)$, there is only one camera C_j's action $a_j(t) \neq a_j(t+1)$ and, its payoff $PO_{C_j}\big(\boldsymbol{a}(t)\big) < PO_{C_j}(\boldsymbol{a}(t+1))$.
Pareto optimal	An outcome of a game is Pareto optimal if there is no other outcome that makes every player at least as well off and at least one player strictly better off [21].
Pareto frontier	A set of strategies which are Pareto optimal. The strategies on the Pareto frontier are equally optimal.

notations in Table 4.1 and the definitions for related terminologies in Table 4.2. In our problem, if at some time instant, there are n_C cameras that can see person P_i, we say that these n_C cameras compete to track this person P_i. Thus, if we let the cameras that can see P_i be the players, then, the case that every time these cameras compete to track P_i can be considered as a multi-player game.

4.2. Game Theoretic Framework

There are only a few existing approaches that solve the camera selection problem in a game theoretic fashion. Two of the most recent game theoretic approaches are the potential game approach introduced in Chapter 3 and [40]. Both the potential game approach and [40] formulate the camera selection problem as a potential game. Instead, in this chapter, we formulate this problem as a *weakly acyclic game*. Weakly acyclic game is a super class of potential games and, thus, it can relax some of the limitations of the potential game model. In Table 4.3, we compare the differences of these two game theoretic models. We can observe that, in a weakly acyclic game, the local payoff function does not have to be aligned with the global one. This provides us much flexibility in the payoff function design. In this chapter (as is shown in the previous section), a video network may have different payoff functions for cameras in different groups. For example in one group of cameras face resolution may be the most important, while in another group of cameras tracking of individuals is more important. Due to this flexibility, we do not have to calculate the global payoff and the system can be distributed to a large extent.

The proposed approach is different from the potential game approach in the

Table 4.3: A comparison of weakly acyclic game and potential game

	Potential game	Weakly acyclic game
Definition	Player action sets $\{\boldsymbol{A}_j\}_{j=1}^{n}$ together with player objective functions $\{U_j: \boldsymbol{A}_j \rightarrow \mathbb{R}\}_{j=1}^{n}$ constitute a potential game if, for some potential function $\phi: \boldsymbol{A}_j \rightarrow \mathbb{R}, U_j\big(a_j''(t), \boldsymbol{a}_{-j}(t)\big) - U_j\big(a_j'(t), \boldsymbol{a}_{-j}(t)\big) = \phi\big(a_j''(t), \boldsymbol{a}_{-j}(t)\big) - \phi\big(a_j'(t) - \boldsymbol{a}_{-j}(t)\big)$.	A game is weakly acyclic if and only if there exists a potential function $\phi: \boldsymbol{A}_j \rightarrow \mathbb{R}$ such that for any action $\boldsymbol{a}(t) \in \boldsymbol{A}_j$ that is not a Nash Equilibrium (NE), there exists a player C_j with an action $a_j{}' \in \boldsymbol{A}_j$ such that $PO_j\big(a_j'(t), \boldsymbol{a}_{-j}(t)\big) > PO_j\big(a_j(t), \boldsymbol{a}_{-j}(t)\big)$ and $\phi\big(a_j{}'(t), \boldsymbol{a}_{-j}(t)\big) > \phi(a_j(t) - \boldsymbol{a}_{-j}(t)$.
Pros	There are a lot of existing learning algorithms [19] for potential games to get the Nash Equilibrium (NE).	It's a superset of potential games. There are fewer limitations (such as the alignment requirement) when designing local utility functions for weakly acyclic games. The local utility function can be time-variant (and, thus, it is a sometimes weakly acyclic game).
Cons	The Nash Equilibrium (NE) in a potential game may not be unique [21]. Thus, it requires learning algorithm to find the NE that is a consensus among all the players. The utility functions must be time-invariant.	There are fewer learning algorithms for weakly acyclic games compared with potential games [21].
Learning algorithms	Action (utility)-based fictitious play; Regret matching; Spatial adaptive play [21].	Payoff-based dynamics; Stochastic better-reply dynamics. [21]
App. in CN[1]	[6][19].	None.

[1] App. in CN stands for Applications in camera networks

following aspects:

1) We model the camera selection and handoff problem as a weakly acyclic game instead of a potential game. By modeling the problem as a weakly acyclic game, we get rid of (a) alignment constraint of local and global utilities and (b) the computation of person utilities. This makes the number of iterations smaller than that for the potential game approach in.

2) We allow different cameras in a network to be in different groups and have different payoff functions while the potential game approach does not. This is also different from [43], where groups are allowed to exchange information only. In our proposed system, different groups of cameras may have different emphases based on the

user's preference.

3) The proposed approach takes handoff smoothness into consideration while the potential game approach does not. This is also different from [43]. In [43], the same kind of information is exchanged among the intra-agency layers to track an object. In our proposed system, we can put different emphases for different groups based on the user's preference. Also, we do not make use of any 3D information.

4) We provide significant new experimental results and proof of convergence in this chapter.

An overview of the proposed payoff based weakly acyclic game approach for camera selection, handoff and active control is given in Figure 4.1.

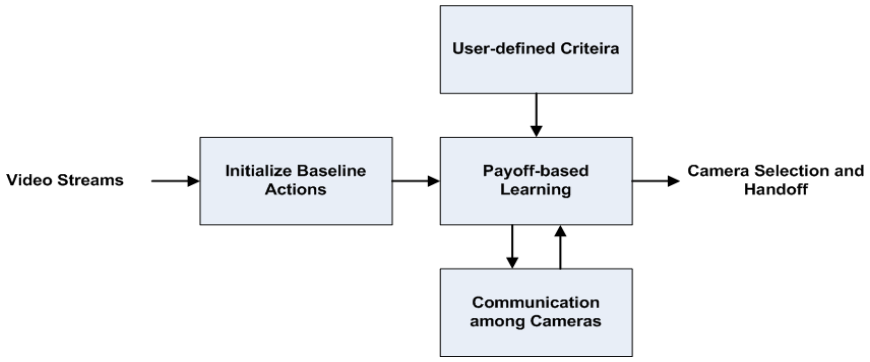


Figure 4.1: Overview of the Proposed Approach

4.3. Weakly Acyclic Game for Camera Selection and Handoff

In this section we discuss the details of the proposed weakly acyclic game approach.

4.3.1. Mapping Camera Selection, Handoff to Weakly Acyclic Game

Let $\boldsymbol{A}_j$ be the set of actions for camera j, then $\boldsymbol{A}_j$={sleep, awake – free of task, awake – recording, awake – recording for P_1,… awake – recording for P_i,…, awake – recording

for $P_{N_P}\}$. At each time instant, the actual action of camera j, $a_j(t)$, may equal to any of the elements in set $\boldsymbol{A}_j$. The camera selection profile is the set of each camera's strategy at the current moment, $\boldsymbol{a}(t) = \{a_1(t), a_2(t), \dots, a_j(t), \dots, a_{n_c-1}(t), a_{n_c}(t)\}$, where $a_j(t)$ is the strategy for camera j at time t. Let $\boldsymbol{a}_{-j}(t) = \{a_1(t), a_2(t), \dots, a_{j-1}(t), a_{j+1}(t), \dots, a_{n_c}(t)\}$ be the strategy profile for all the cameras other than camera j at time t. A game G is said to weakly acyclic if there exists a better reply path (the system will gain more payoff by taking strategies along this path) starting at some strategy profile and ending at some pure Nash Equilibrium of G. A better reply path is a sequence of camera action profiles $\boldsymbol{a}(1), \boldsymbol{a}(2), \dots, \boldsymbol{a}(T)$ such that for each successive pair $(\boldsymbol{a}(t), \boldsymbol{a}(t+1))$ there is exactly one player such that $a_j(t) \neq a_j(t+1)$ and for that player $PO_{C_j}(\boldsymbol{a}(t+1)) > PO_{C_j}(\boldsymbol{a}(t))$ [42]. Since we can change the camera actions at each iteration of the learning process (see the payoff-based learning in the next section), we can always find a better reply path for the camera action assignment until it reaches the optimal Nash Equilibrium. So, this process can be modeled as a weakly

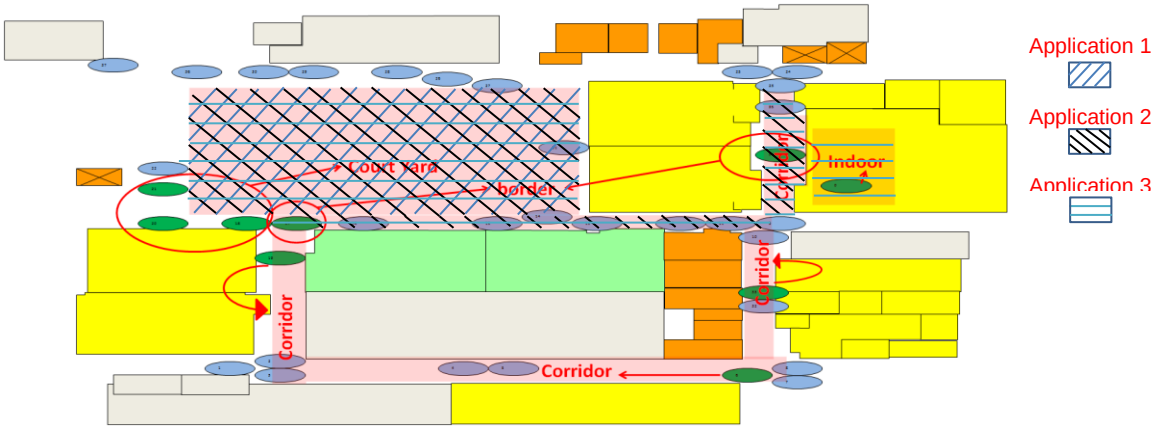


Figure 4.2: Map of the VideoWeb camera network [50]. The green ones are the cameras used in our experiments. The blue ones are those not used. The light red regions are the regions where the experiments are carried out.

acyclic game. Since the game is finite and the path cannot cycle back on itself according to the definition, the last reply path has to be the optimal solution of the camera selection.

4.3.2. Payoff-based Learning Algorithm

In the following, we describe a payoff based learning algorithm for weakly acyclic game that helps us to find the optimal NE.

First, we will group our cameras into 4 categories: G1 = indoor; G2 = court yard; G3 = corridor; G4 = border. The border cameras are awake all the time, once an object is detected by the border camera, the border camera will awaken its neighboring cameras. Since a person's trajectories are not known *a priori*, no prior knowledge is available to estimate $\boldsymbol{A}_j$'s distribution. We set $\boldsymbol{A}_j$ of the border cameras as awake at all times, and all the other cameras' $\boldsymbol{A}_j$ are set to sleep by default because of the power consumption consideration, which is very normal in real-world video surveillance systems. Then, for all the awaken cameras we run the payoff-based learning algorithm:

(1) Initialization: we first randomly choose a baseline action and baseline payoff for each awake camera.

$$a_j^b(1) = a_j^b(0)$$

$$PO_j^b(1) = PO_j(0)$$

Calculate $PO_{C_j}(\mathbf{0})$ using equation (6);

(2) Exploration: At each learning iteration step *k*,

$a_j(k)$ is randomly selected from $\boldsymbol{A}_j \backslash \{a_j(k-1)\}$ with probability ε.

$a_j(k) := a_j^b(k-1)$ with probability $(1-\varepsilon)$.

(3) Update: If the result of (2) is a better reply path, then we replace the baseline actions with this better reply path, otherwise, we keep the baseline actions to be the same as in the previous step.

if $(a_j(k)$!= $a_j(k-1))$

if ($PO_{C_j}(\boldsymbol{a}(k)) < PO_{C_j}(\boldsymbol{a}(k-1)) + \delta)$

$a_j^b(k) = a_j(k-1);$

$PO_{C_j}^b(k) = PO_{C_j}(a(k-1));$

else

$a_j^b(k) = a_j^b(k-1);$

$PO_{C_j}^b(k) = P_{C_j}^{Ob}(k-1);$

else

$a_j^b(k) = a_j^b(k-1);$

$PO_{C_j}^b(k) = PO_{C_j}(k-1);$

A better reply path in our problem is a set of camera action assignment that can make the camera which is changing its strategy to have a higher payoff than that at the previous iteration step. By performing this at each iteration, we will finally get a better reply path which makes sure that each camera gains the maximum payoff. The exploration rate ε is any real number belonging to [0, 1]. The trade-off for choosing ε is in Section 4.4.1.

In our work, the calculation of the payoff for each camera is localized to the camera node itself. The only information that might be needed from other cameras is the

ALGORITHM 1. CAMERA SELECTION AND HANDOFF BY USING WEAKLY ACYCLIC GAME
This algorithm learns the better reply path during the camera selection process. We get the camera assignment for the current frame when the algorithm terminates.
Input: Each camera's available action set $\boldsymbol{A}_j$.
Output: Each camera's action for the current frame.
Parameters: current frame image current_frame, camera *j*'s available action set $\boldsymbol{A}_j$, exploration rate ε, improvement step δ, number of iterations N_{iter}.

```
Procedure CameraSelection{
   GrabProp (current_frame);
      for i := 1:N_P
         for j := 1:N_C
            if (IsDetected (i, j))
                  Initialize according to step (1);
         for k := 1: N_iter
             Explore according to step (2);
             Update according to step (3);
         If (IsMultiple(i))
             a_i = arg max_{j=1:N_C} PO_j
}
Procedure IsDetected (i, j){
   if person P_i is detected by camera C_j
      return 1;
   else
      return 0;
}
Procedure IsMutiple (i){
   if person P_i is selected by multiple cameras
      return 1;
   else
      return 0;
}
```

status (action $\boldsymbol{A}_j$) of its neighbor, which is broadcast within the group. Even if when there is only one group of cameras, i.e., the information is broadcast among all the cameras, this does not require much bandwidth or any centralized computation, since all the computation for each camera's payoff does not need any centralized control. This information exchange is performed before the payoff-based learning algorithm starts. During the process of payoff-based learning, there is no exchange of any information. After the learning process, when there are multiple cameras select the same person, the system will use the constraint that a person can be tracked by one camera only to select

the “best” camera and filter out the others. The overall algorithm is given in Algorithm 1.

4.3.3. Design of Criteria

The criteria used for the proposed approach should be user supplied. In this chapter, we proposed five criteria, $\{S_{im}\}|_{m=1}^{5}$ to denote the criteria for person P_i. The reason to have these criteria is that these are properties demanded by a video surveillance system in most cases, as discussed below:

1) Time Delay. It takes time for a camera to change from the sleep mode to the awake mode. If a camera was in the sleep mode and to be awakened to follow an object, there is a possibility that the object has moved out of the FOV of this camera during its being wakened up. So, we give some penalty for wakening up a camera.

$$S_{i1} = \begin{cases} 0, & if\ a_j(t-1) = sleep\ and\ a_j(t) = awake \\ 1, & otherwise \end{cases} \quad (1)$$

2) Position of a Person. We prefer the camera with the object near to its center, because there is least distortion and it allows enough time to switch between cameras. If the camera handoff is taken only when an object is leaving the FOV of a camera, the delay caused by the awakening process may lead to the loss of the object for several frame. This criterion is measured by the Euclidean distance between the centroid of a person and the center of the camera image.

$$S_{i2} = \frac{\sqrt{(x-x_c)^2+(y-y_c)^2}}{\frac{1}{2}\sqrt{x_c^2+y_c^2}} \quad (2)$$

where (x, y) is the current position of the person and (x_c, y_c) is the center of the image plane. We suppose that the farthest distance between an object and the center of the image is half the length of the diagonal.

3) Size of a Person. In most video surveillance scenarios, an object with too small or too large size is hard to be observed. We have this criterion to choose the object with a proper size when he/she is visible in more than one camera. This criterion is measured by the ratio of the number of pixels inside the minimum bounding box of a person to that of the size of the image plane. Here, we assume that neither a too large nor a too small object is convenient for observation. Assume that λ is the threshold for best observation, i.e. when $r = \lambda$ this criterion reaches its peak value, where $r = \frac{\#\ of\ pixels\ inside\ the\ bounding\ box}{\#\ of\ pixels\ in\ the\ image\ plane}$.

$$S_{i3} = \begin{cases} \frac{1}{\lambda} r, & when\ r < \lambda \\ \frac{1-r}{1-\lambda}, & when\ r \geq \lambda \end{cases} \tag{3}$$

4) View of a Person. In some cases, a person's frontal view is preferred because it can provide more features of interest. So in some of the experimental cases, for some groups of cameras, we have this criterion to emphasize on the frontal view if it is visible. Similar to the previous criterion, we assume that the threshold for best frontal view is γ, i.e. when $R = \gamma$ the view of the person is the best, where $R = \frac{\#\ of\ pixels\ on\ the\ face}{\#\ of\ pixels\ on\ the\ entire\ body}$.

$$S_{i4} = \begin{cases} \frac{1}{\gamma} R, & when\ R < \gamma \\ \frac{1-R}{1-\gamma}, & when\ R \geq \gamma \end{cases} \tag{4}$$

5) Smoothness. In most of the previous work [14][8][39][37], the authors do not consider how to handoff from one camera to another smoothly. As a result, when it is hard to decide which camera to use, the system may keep handing over among multiple cameras, which is not desired. This criterion avoids oscillations when doing handoffs between two cameras too frequently. This is the case when the payoffs of two cameras are quite similar. (Note that it has nothing to do with the trajectory smoothness).

$$S_{i5} = \begin{cases} 1, & if\ D > 0 \\ 0, & otherwise \end{cases} \tag{5}$$

$$D = \sum_{m=1}^{4} w_{mC} S_{imC} - \frac{1}{C-K} \sum_{f=K}^{C-1} \left(\sum_{m=1}^{4} w_{mfi} S_{mfi} \right)^{\left(1 - \frac{1}{f-K+2}\right)}, C > K$$

where K is the last frame number from where the current camera is used to track P_i, C is the current frame number. D evaluates how well (based on the previous criteria) the current camera can be used to continue tracking person P_i. The first term of D is the payoff that the current camera can g et from tracking P_i. The second term is the average payoff this camera gets from tracking P_i since the last time it is used for this person. But we make the importance of the payoffs from the previous frames decrease exponentially, i.e., the most current frame has the highest importance for deciding the average value.

The payoff function of a camera is defined as the weighted sum of the above five criteria (Equations (1)-(5)) for all the persons that can be seen by it.

$$PO_{C_j} = \sum_{i=1}^{n_P} \sum_{m=1}^{5} T_i w_m S_{im} \tag{6}$$

where $T_i = \begin{cases} 1, if\ C_j\ is\ assigned\ to\ track\ P_i \\ 0, \qquad\qquad\qquad otherwise \end{cases}$.

It should be noticed that all these criteria are just an example of those that can be applied in the proposed system. We use these criteria in our experiments, but the design of criteria should reflect the user's preference and one should not be limited to these criteria. If the criteria input by the user are different, it means that the user has different metrics on deciding when to do the camera selection and handoff and, of course, the overall result will change accordingly.

4.3.4. Convergence, Scalability and Optimality of the Algorithm

Statement 1—Given a small enough ε and a large number of iterations we will reach an optimal Nash Equilibrium with an arbitrarily high probability [42].

Let $\boldsymbol{a}^b(\mathrm{t})$ be the cameras' baseline action vector at time *t*, $\boldsymbol{p}^b(t)$ be the baseline camera payoff vector at time *t*. Then $[\boldsymbol{a}^b(t), \boldsymbol{p}^b(\mathrm{t})]$ denotes the dynamic state during the camera selection process. There are three possible sets of $[\boldsymbol{a}^b(t), \boldsymbol{p}^b(\mathrm{t})]$ which are defined as given below:

A: $p_i^b(t) \neq PO_{C_j}(\boldsymbol{a}(t))$ for at least one Camera C_j.

B: $p_i^b(t) = PO_{C_j}(\boldsymbol{a}(t))$ for all the cameras and $\boldsymbol{a}(t)$ is a Nash Equilibrium.

C: $p_i^b(t) = PO_{C_j}(\boldsymbol{a}(t))$ for all the cameras but $\boldsymbol{a}(t)$ is not a Nash Equilibrium.

A Nash Equilibrium in our problem is a set of camera assignment $[\boldsymbol{a}^*, \boldsymbol{p}^*]$ such that no camera can achieve higher utility by deviating from this assignment.

To prove statement 1, we need the following facts [42]:

1) Any state $[\boldsymbol{a}^b(t), \boldsymbol{p}^b(t)] \in A$ transitions to a state in $B \cup C$ in one period with probability *O(1)*.

2) Any state $[\boldsymbol{a}^b(t), \boldsymbol{p}^b(t)] \in B \cup C$ transitions to a different state with probability at most $O(\varepsilon)$.

3) For any state $[\boldsymbol{a}^b(t), \boldsymbol{p}^b(t)] \in C$, there is a finite path to $[\boldsymbol{a}^*, \boldsymbol{p}^*]$, with each transition with probability $O(\varepsilon)$.

4) For any equilibrium $[\boldsymbol{a}^*, \boldsymbol{p}^*] \in$ B, any path that deviates from this and does not loop back to $[\boldsymbol{a}^*, \boldsymbol{p}^*]$ has the probability of $O(\varepsilon^k), k \geq 1$ for each transition step in the path.

The proof involves implementing the resistance tree data structure to describe games. We only describe it here briefly. The interested reader is referred to [42] for further details.

For every possible state, we can:

1) Construct minimum resistance trees with these states as vertices.

2) Decrease the resistance of a tree by replacing a path with another one with lower resistance, such as putting in a path with probability $O(\varepsilon)$ and subtract a path with probability of *O(1)*, which can be regarded as a better reply path in our context.

3) Finally, we will get the trees rooted at the Nash Equilibrium to have the lowest resistance.

Experimental results show that the convergence (see Figure 4.3(b)) is reached pretty fast although there is no guarantee that within so many iterations one can get the optimal Nash Equilibrium.

4.4. Experiments

4.4.1. Datasets and Parameters

A. Datasets

We use the V*ideoWeb* camera network that consists of 37 outdoor Axis 215 cameras [50]. A floor map of these cameras is shown in Figure 4.2 where the camera groups are marked with red circles. These are the regions where the actual experiments are conducted. We perform experiments with different numbers of cameras and persons. (For indoor experiments, we have an extra cameras placed in our lab which is not shown in the map.) All the persons walk in a natural manner. The data is acquired at 10 frames per second. The videos from different cameras are synchronized by the time stamps shown in the file name. We save the videos in .jpg files with the machine time as the file names. The precision for synchronization is ~ 0.33 second. For the readers' convenience, we provide a summary of different experimental cases in Table 4.4.

To remove the effect of different trackers on camera selection and handoff [51], we do all the experiments in two ways: (1) Use annotated videos, i.e., the ground-truth data for tracking; (2) Use the particle filter tracker [52] when no annotation are provided.

Table 4.4: Different cases for the experiments in Chapter 4

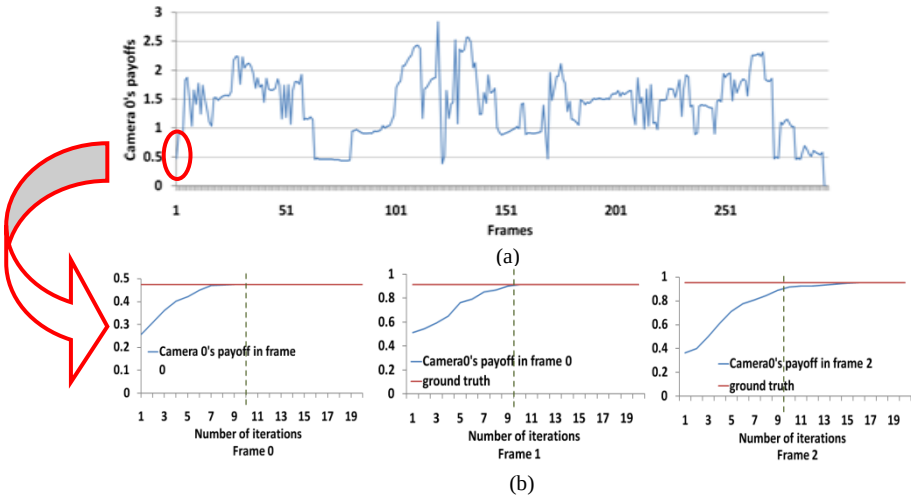
Case #	N_P	N_C	Id	Cd	Od	N_F
Case 1	6	4			√	297
Case 2	5	4			√	712
Case 3	5	3			√	1016
Case 4	6	3		√	√	1329
Case 5	8	4		√	√	1728
Case 6	8	6	√	√	√	2238

N_P: *number of persons*; N_C: *number of cameras*; Id: *indoor*;
Cd: *corridor*; Od: *outdoor*; N_F: *number of frames*

B. Parameters

There are 4 key parameters: the number of iterations N_{iter}, the exploration rate ε, the improvement step δ and the weights for different criteria w_m.

1) *Number of iterations* N_{iter}: According to the discussion in the Section 4.3.4, we can get the optimal camera selection solution with any small probability when $N_{iter} \to \infty$. In our experiments, we test on some sample frames with a relatively large N_{iter} and select the number for which over 99% of the frames can get the optimal value. We show some example frames (Frame 0 to Frame 2) by applying 20 iterations in Figure 4.3(b). This shows how the camera payoff values can converge by finding a better reply path. As we can see, after 10 iterations, the payoff derived by the proposed approach is equal to or very close to the ground-truth. So, in our experiments, we use 10 iterations for each frame. The corresponding camera payoffs by using 10 iterations and the ground-truth values are shown in Figure 4.3(a) (Camera 0 is used in this case).



(a)

(b)

Figure 4.3. (a) Camera 0's payoffs in each frame. (b) Camera 0's payoffs from iteration 1 to iteration 20 in frame 0 to frame 2.The green dashed line is where we cut the number of iteration and use it for each frame in this experiment as shown in (a).

2) The exploration rate ε: ε controls the speed with which the learning algorithm converges. The larger the value of ε, the faster the algorithm converges. But the tradeoff is that the higher the value of ε, the higher the chance that the algorithm will miss the optimal value. ε is set to be 0.1 in our experiment, i.e., in each iteration, each camera changes its action with a probability of 0.1. This is a reasonable value because we test $\varepsilon = 0.2$ and in that case there are 15% of the frames which miss the optimal value. With $\varepsilon = 0.1$, the results are shown in Table 4.5 to Table 4.7.

3) The improvement step δ: It accelerates the algorithm to reach the optimal value. Similarly to the value of ε, if δ is set too large, the possibility of missing the optimal value will be high. We test δ from $\delta = 0.005$ to $\delta = 0.02$ with a step of 0.005, and select the value of $\delta = 0.01$ empirically. The results using this value are shown in Table 4.5 to Table 4.7.

4) The weights for different criteria w_m: Since these weights imply the user's preference of one criterion over another, the values of w_m has to be provided by the user and are normalized to [0,1] before the entire system can run. In our experiments, we have different sets of values for w_m. This is because in some applications, we favor on an object's frontal view while in some other applications we do not.

The total number of people is estimated as below:

1) Detect humans in all the cameras [56].

2) Perform an initial match of individuals using homography.

3) If two or more persons are matched to be the same person in different cameras using homography. Then compute this correlation using both color histogram and texture

features (such as local binary patterns). If the value is below a fixed threshold, conclude that the two persons are not the same. The threshold for the correlation coefficient is trained using the first 10% frames of a video sequence (30, 71, 102, 133, 173, 224 frames for the 6 cases listed in Table 4.4, respectively). This has to be done to every video sequence because we have different camera network setups in different experimental cases. This is like an onsite parameter tuning when installing cameras, which is very normal in nowadays video surveillance systems [54]. The selected correlation coefficients are 0.41, 0.41. 0.39, 0.45, 0.45, and 0.45, respectively for different cases shown in Table 4.4.

4) If a person is occluded in one camera but visible in another, the approach in step (3) will detect a person and add this person to the total count of the number of people in the system. If a person is occluded in all the cameras, this person is not added to the total count.

5) Register all the detected persons in the system. When a person enters the system, repeat step (3) - (4). If this person is matched to a registered person, then this person is not added to the total count, otherwise, the total count is updated.

4.4.2. Performance Metrics

The goal is to have an automated system to select a “best” camera of each person at each time instant. The “best” camera has different meanings in different applications. In cases, such as following a person, the only requirement is that this person is not lost as long as the person is still in the coverage of the camera network. In low resolution videos, we can only identify persons by their shapes or colors of clothes and details on faces are not

available. In such cases, we are concerned with tracking a person smoothly in the network while frontal view of a person is not important. On the contrary, in some applications, we have enough resolution to see a face clearly such that we can use this information to identify a person. Thus, in these cases, the camera with a person's frontal face is preferable.

We will show experiments for three different application scenarios:

Application 1. Tracking people in a camera network smoothly.

Application 2. Tracking people in a camera network smoothly and frontal face is preferable.

Application 3. Tracking people in a camera network smoothly and frontal face is preferable only in a selected area.

Accordingly, we have different error definitions for different applications. We use Err_1, Err_2, Err_3 to denote the errors for the above three applications, respectively. In *Application 1*, we say there is an error if a person is in the coverage of the camera network but no camera is selected to follow this person. We also want to follow a person smoothly, meaning that the camera selection does not dither among multiple cameras. So, if there are more than 3 handoffs in 10 frames (i.e., in 1 second), we consider there is a *dithering error*. Dithering may only happen when there are two or more cameras whose tracking performance are almost the same. This kind of situation does not occur frequently in a video, since in a video network, we want to use multiple cameras to cover an area as large as possible to make full use of each camera. Thus, it is normal that the overlaps between multiple cameras' views are not too large. Also, there are tradeoffs

between the dithering rate and the camera selection quality. If the dithering rate is high, the camera selection quality can be guaranteed but the high frequency switches among cameras are not desired for visual observations. On the other hand, if the dithering rate is set too low, the system will have more emphasis on the tracking smoothness and, thus, will miss the camera with a better tracking performance. We use number 3 as the dithering rate threshold in our experiments, because in most cases, this number can give good tradeoff between the observation data and camera selection quality. In *Application 2*, since frontal face is preferred, besides the smoothness error, we will consider it as an error if a person's frontal face is available in camera C_j but C_j is not selected to track this person. *Application 3* is a combination of the above 2 cases. We will have some cameras in a group looking for persons' frontal views in a selected area, while other cameras in other groups only care for tracking persons smoothly.

We define the following terms:

S_{lost}: The set of frames such that a person is visible in the camera network but is lost by the system.

S_d: The set of frames such that there is a dithering error in a video sequence. By dithering, we mean the camera handoff dithering when the hand-off takes place more than a predefined number of times per second.

S_{fl}: The set of frames such that a person's frontal view is available in at least one camera, but no frontal view is selected by the system.

S_{gfl}: This has the same meaning as S_{fl}, but only within the group of cameras where frontal view is preferred.

$Num\{S\}$: The number of elements in set *S*.

N_F: The number of frames of one camera. In our experiments, of all the cameras are the same.

Thus,

$$Err_1 = Num\{S_{lost} \cup S_d\}$$

$$Err_2 = Num\{S_{lost} \cup S_d \cup S_{fl}\}$$

$$Err_3 = Num\{S_{lost} \cup S_d \cup S_{gfl}\}$$

Note that in the above equations we use ∪ instead of +, because there are some frames containing more than one error, i.e., the system may lose one person and lose the frontal view of another person in its FOV. In this case, we only count the error once for such a frame. So, when computing the total error rate ER_i, we can divide the number of errors by the number of frames. That is

$ER_i = \frac{Err_i}{N_F \times N_C}, i = 1, 2, 3,$

where N_F and N_C are number of frames in video and number of cameras, respectively.

To calculate these errors, we need the ground-truth data. There are two kinds of ground-truth. 1) Ground-truth for tracking. We generate this ground-truth using an online available tool ViPER-GT [53]. This is done manually. 2) Camera selection ground-truth. The cameras are fixed in this chapter and there is no active control of cameras. We exhaustively enumerate all the possible actions that a camera can take, calculate the payoffs according to the predefined criteria (Section 4.3.3) and choose the best result as the camera selection ground-truth. Different approaches have different principles for

doing camera selection. For example, the fuzzy-based approach [25] does camera handoff based on objects' location only. This is different from the proposed approach where we focus on multiple user-supplied criteria. We use the camera selection ground-truth when comparing different approaches only to show under this scenario, i.e., with these user-supplied criteria, how different approaches perform.

4.4.3. Approaches Compared

A. The Potential Game Approach

In a potential game, we have to use the same utility (payoff) design for all the cameras, which are all aligned with the global utility function. In order to achieve this alignment, person utilities have to be computed from which we can get the camera utilities (See Chapter 3 for more details).

B. The Fuzzy-based Approach [25]

To demonstrate the advantage of the proposed approach over the other ones, we perform comparison with another non-game theoretic approach [25], the fuzzy-based approach. The fuzzy-based approach is a decentralized approach. In this approach, each candidate camera has two states for the object that is in its FOV: the non-selected state and the selected state for tracking. Then, camera handoff is done based on the camera's previous state S_i and the tracking level state SS_i, which is defined by estimating the position measurement error in the monitoring area. The two states for the tracking level are: unacceptable, meaning that the object is too far away and acceptable, meaning that the object is within the FOV and the quality is acceptable. The tracking level state can be decided by the proposed Criterion 2, S_{i2}.

4.4.4. Experiments for Different Applications

In this section, we provide experimental results to corroborate the theoretical approach proposed in Section 4.2. All the experiments are done in the real-life video network content.

A. Experiment #1: Tracking people across camera network – Application 1 (Case 1 and 2 in Table 4.4)

In this experiment, we will show how the proposed approach can track a number of persons smoothly. There is only one group of cameras and we are not concerned with frontal view in this experiment, so w_1 and w_4 are set to 0. The other criteria weights are $w_2 = 0.2, w_3 = 0.3, w_5 = 0.5$.

We show example frames by the proposed approach in Figure 4.4(a) and Figure 4.4(b). The results shown here are for a single person only for easy observation. We can

Figure 4.4: Experiment #1. Example frames for comparing smoothness during camera handoffs. The left column ((a) and (c)) corresponds to the left red ellipse in Figure 4.5 while the right column ((b) and (d)) corresponds to the right red ellipse in Figure 4.5. The first row is the result by using the proposed weakly acyclic game (WAG) approach while the second row is the result by using the potential game approach in Chapter 3. For easy observation, we only draw the bounding box for the person under discussion in the text. Person 4: the left column. Person 5: the right column. The red number at the left bottom is the ID of the camera that is assigned to track the person.

notice that in Figure 4.4(a), the system hands over from Camera 0 to Camera 2 to achieve a better size and position of the person in blue bounding box (Person 4 in Figure 4.5). Whereas in another case, in Figure 4.4(b), the size and position of the person in brown bounding box (Person 5 in Figure 4.5) are both good according to the criteria, and, thus, no camera handoff takes place, even if this person is also visible in other cameras (See Figure 4.4(d)). This shows that with the smoothness criterion, S_{i5}, being considered, camera handoff happens only when it is necessary to avoid dithering among different cameras. The overall camera selection results by the proposed approach for Case 1 is shown in Figure 4.5(a). We can notice that there are no frequent camera transitions (more than 3 camera handoffs within 10 frames) by the proposed approach.

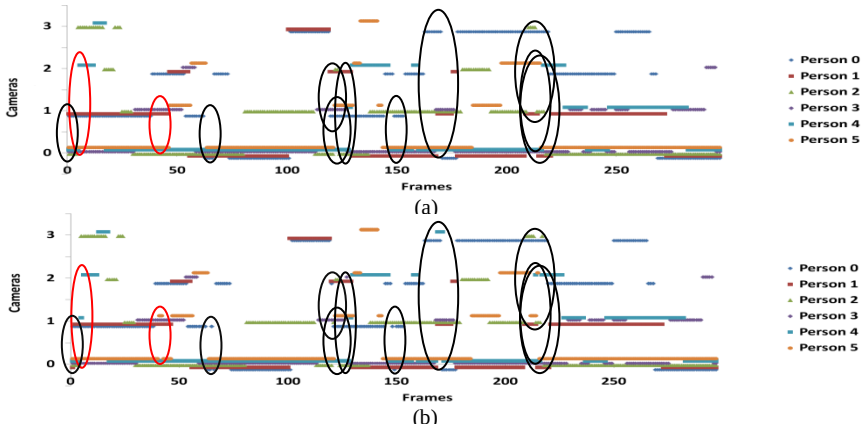


Figure 4.5: Camera selection and handoff results for each frame in Experiment #1. Different colors are used for the camera selection and handoff results for different persons. When the same camera is selected for different persons, we draw the selection and handoff curves around that camera number while keeping the curves apart from one another a little bit to make the figure clear. (a) is the camera selection and handoff results when considering handoff smoothness. (b) is the camera selection and handoff results without considering handoff smoothness, by using the potential game approach. Circled arrears in (b) are unstable states by using the potential game approach. The corresponding frames are circled in (a) for comparison.

For better performance estimation, we do another experiment, Case 2, with similar camera settings, but more frames. In Table 4.5, we list the number of camera handoffs, number of dithering and the error rates for both Case 1 and Case 2. To remove the influence of different trackers on the camera selection approach, we show the result using annotated video and a particle filter tracker, respectively. We also list the S_{lost} and S_d. But it should be noticed that the total number errors are not the summation but union of these two numbers. This is because in some frames, these two errors can appear simultaneously. When calculating the final error rate, we use the union of these two numbers (if a frame shows both of these two errors, it is counted as an error frame only once) as the total number errors. We can observe that in the results with annotated video, where we assume there is no tracking error, the camera selection and handoff error by the proposed approach is less than 2% in the short video case and less than 10% in the longer video case.

For comparison, we show corresponding camera selection results by using the potential game approach in Figure 4.5(b). Since no smoothness is taken into account by this approach, there are many camera transitions shown in this Figure. For example, for

Table 4.5: Comparison of the proposed weakly acyclic game approach and the potential game approach

Case (See Table 4.4	Approach	NH		ND		Error rate						N_F
		AV	PF	AV	PF	AV			PF			
						S_{lost}	S_d	ER_1	S_{lost}	S_d	ER_1	
Case 1	WAG	86	84 (5)	0	1 (1)	0	3	1.01%	17 (1)	8 (1)	7.74% (0.67%)	297
	PG [6]	103	109 (5)	15	18 (3)	15	8	6.73%	29 (3)	13 (2)	12.79% (1.01%)	
Case 2	WAG	159	162 (7)	2	3 (1)	8	6	1.97%	57 (7)	12 (6)	8.99% (1.12%)	712
	PG [6]	192	197 (7)	33	40 (4)	42	23	8.57%	73 (8)	49 (6)	15.17% (1.68)	

NH: number of handoffs; ND: number of dithering among frames; AV: Annotated video; PF: Particle filter tracking; WAG: weakly acyclic game; PG: potential game; the numbers shown for the results with PF are in the form of average (standard deviation)

Person 4, from frame 4 to frame 6, the system hands off from Camera 0 to Camera 1 and then to Camera 2. This result is due to considering the size and position of the person only (see Figure 4.4(c)). But if we take smoothness into account, the handoff from Camera 0 to Camera 1 is not necessary. Similarly, in Frame 40 to Frame 43, for Person 5, in Figure 4.5(b), the system hands off from Camera 0 to Camera 1 and then hands off back from Camera 1 to Camera 0. On the contrary, we can notice that Figure 4.5(a) sticks to Camera 0 during these frames when the quality of Person 5 is still acceptable (see Figure 4.4(b)). The system finally hands off from Camera 0 to Camera 1 when Person 5 leaves the FOV of Camera 0 and enters that of Camera 1, as shown in Figure 4.5(b). These two cases are marked with the red ellipses in Figure 4.5. The corresponding images are shown in Figure 4(d) and Figure 4.4(d). The other unstable states in Figure 4.5(b) are marked in black circles. The corresponding frames in Figure 4.5(a) are also marked for comparison.

In Table 4.5, due to the computational cost, within limited number of iterations, there can be the case when the potential game approach loses the person but finds him/her again in the next frame. Thus, S_d and S_{lost} are highly dependent in this case. Table 4.5 shows that, in both cases, the weakly acyclic game approach outperforms the potential game approach and has fewer handoffs.

B. Experiment #2: People across camera network with frontal faces preferred – Application 2 (Case3, 4, and 5 in Table 4.4)

In Application 2, a user prefers to see a person's frontal view when it is available. So, we switch the value for w_4 from 0 to 0.4. The normalized criteria weights for this application are $w_1 = 0, w_2 = 0.1, w_3 = 0.2, w_4 = 0.4, w_5 = 0.3$.

We show example frames by the proposed approach in Case 3, Case 4 and Case 5 in Figure 4.5(a), (c), (e), respectively. As can be noticed, when a person's frontal view is available in a camera, the system shows its favor to that camera. For example, in Figure 4.6(a) Case 3 Frame 426, the proposed approach selects Camera 2 for the person in red bounding box. Similarly, in Case 3 Frame 574, the proposed approach successfully selects the frontal view for the person in blue bounding box.

In Figure 4.7, we plot the final payoffs (utilities) obtained for Camera 0 in each frame for Case 3. It shows that, the payoff values calculated by the proposed approach are almost aligned with the ground-truth values. This means the error rates by the proposed approach are low. The concrete numbers are shown in Table 4.6.

We also compare our results with the potential game approach. We can notice that the proposed approach get the favorable camera in most cases while the potential game

Table 4.6: Comparison of the proposed weakly acyclic game approach and the potential game approach

Case (See Table 4.4	Approach	NH		ND		Error rate							N_F
		AV	PF	AV	PF	AV			PF				
						S_{lost}	S_d	ER_2	S_{lost}	S_d	S_{fl}	ER_2	
Case 3	WAG	172	184 (5)	5	6 (1)	25	11	3.15%	84 (8)	13 (2)	12 (1)	9.84% (0.98%)	1016
	PG [6]	196	213 (5)	35	38 (3)	48	29	7.09%	101 (8)	64 (4)	33 (2)	15.56% (0.89%)	
Case 4	WAG	199	190 (7)	7	7 (1)	38	15	3.99%	137 (9)	10 (2)	21 (3)	10.99% (0.83%)	1329
	PG [6]	239	222 (6)	39	42 (2)	88	37	10.31%	197 (7)	77 (4)	64 (3)	19.19% (0.83%)	
Case 5	WAG	221	230 (5)	7	9 (1)	50	13	4.11%	186 (8)	14 (2)	13 (1)	11.92% (0.64%)	1728
	PG [6]	267	278 (6)	48	55 (4)	132	44	9.61%	309 (9)	132 (4)	46 (2)	22.51% (0.75)	

NH: number of handoffs; ND: number of dithering among frames; AV: Annotated video; PF: Particle filter tracking; WAG: weakly acyclic game; PG: potential game; the numbers shown for the results with PF are in the form of average (standard deviation)

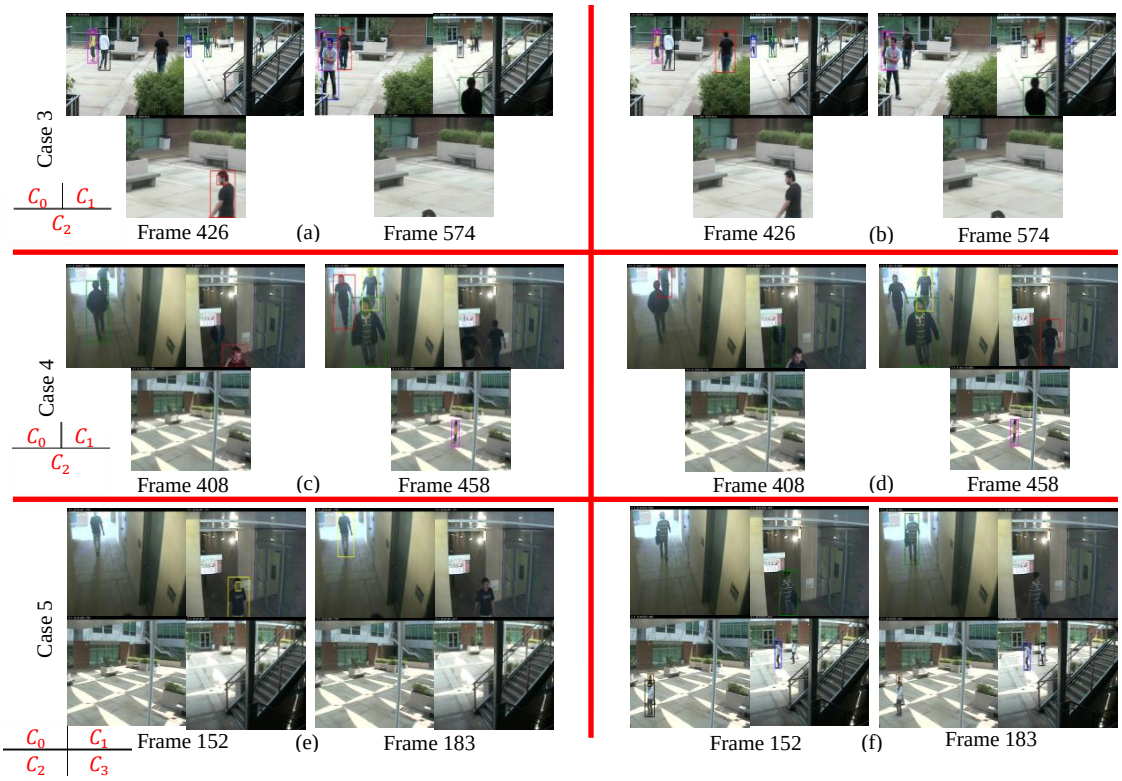


Figure 4.6: Experiment #2. Example frames for Case 3, Case 4 and Case 5, for comparison of the weakly acyclic game approach (the left column) and the potential game approach (the right column). These are some of the example frames in which the proposed weakly acyclic game approach is able to pick the camera with frontal face of a person while the potential game approach fails to do so. The red number is the Camera ID.

approach may lose the best camera because of the lack of available iterations. For example, in Figure 4.6(a) Case 3 Frame 426, the potential game approach loses the frontal view for the person in red bounding box and selects Camera 1. In Case 3 Frame 574, the potential game approach loses the frontal view for the person in blue bounding box. The utilities obtained for Camera 0 in each frame for Case 3 by using the potential game approach is shown in Figure 4.7 as well. There are larger variances from the ground-truth value by using the potential game approach as compared to the proposed approach.

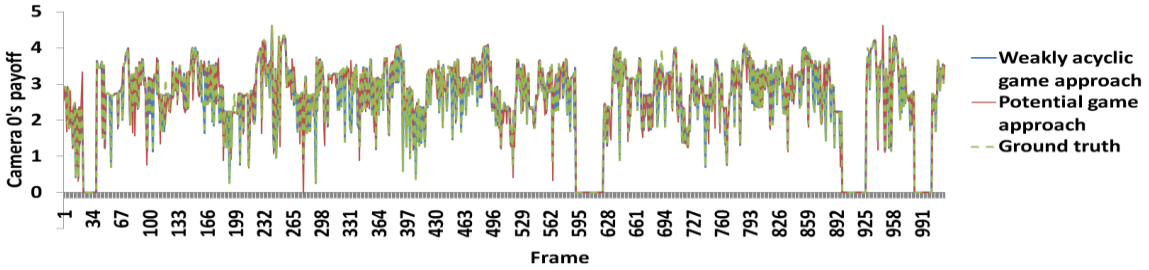


Figure 4.7: Camera 0′s payoff in each frame by using the proposed approach and the potential game approach in Case 3. The weakly acyclic game approach deviates less from the ground-truth as compared to the potential game approach.

The overall results for the two compared approaches are summarized in Table 4.6. There are always fewer handoffs by the weakly acyclic game approach because it considers smoothness when switching among the cameras. The weakly acyclic game approach also has a lower error rate, especially in more complicated cases (more people, more cameras), because it can reach the optimal solution within a fewer number of iterations. The experiments with the particle filter tracker have a higher error rate than those with annotated videos because the camera payoffs, which are based on the tracking quality, are related to tracking errors.

After comparing the proposed approach with the potential game approach in the above two applications, we conclude that the weakly acyclic game approach outperforms the potential game approach because of the following reasons:

1) Smoothness is considered as one of the criteria. This is implied by the dithering error, S_d.

2) The computational efficiency. Because there is no requirement for local payoff to be aligned with the global payoff, we do compute the person payoff and global payoff in this chapter. So, given the same small number of iterations, when the weakly acyclic

game reaches the optimal solution, the potential game approach may not. This is implied by the number of frames that a person is lost (S_{lost}) or the preferred frontal view is lost (S_{fl}). Note that in this case, the frontal view is also a criterion in Chapter 3, but S_{fl} in the potential game approach are still higher than those in the proposed weakly acyclic game approach.

3) Because there is no alignment requirement, it will be easier for the proposed approach to include more criteria than the potential game approach.

C. Experiment #3: Tracking people across camera network (different groups of cameras have different criteria) – Application 3 (Case 6 in Table 4.4)

Unlike the previous experiments where cameras belonging to the same group are used, in this experiment, in this experiment, we use 2 cameras at the court yard, 3 cameras in the corridor and another indoor camera. This is done to simulate different groups of cameras. In addition, the corridor camera and one of the court yard cameras are set as the border camera and will be on all the time during the experiment. Hence, the weights for S_{i1} is changed from 0 to 0.1 in this case and all the other weights are normalized accordingly. We acquire frontal view from the indoor and corridor cameras only and for all the other cameras we need smooth tracking and handoffs only. The locations of different groups are marked in Figure 2. The camera FOVs are shown in Figure 4.8.

This is the most complex experiment presented in this chapter. It has the maximum number of cameras (6) and the maximum number of persons (8). It covers the largest area (from indoor to corridor to outdoor, ~ 2000 square feet) involves different groups of cameras with different criteria applied. There 2238 frames in this video. Here we also

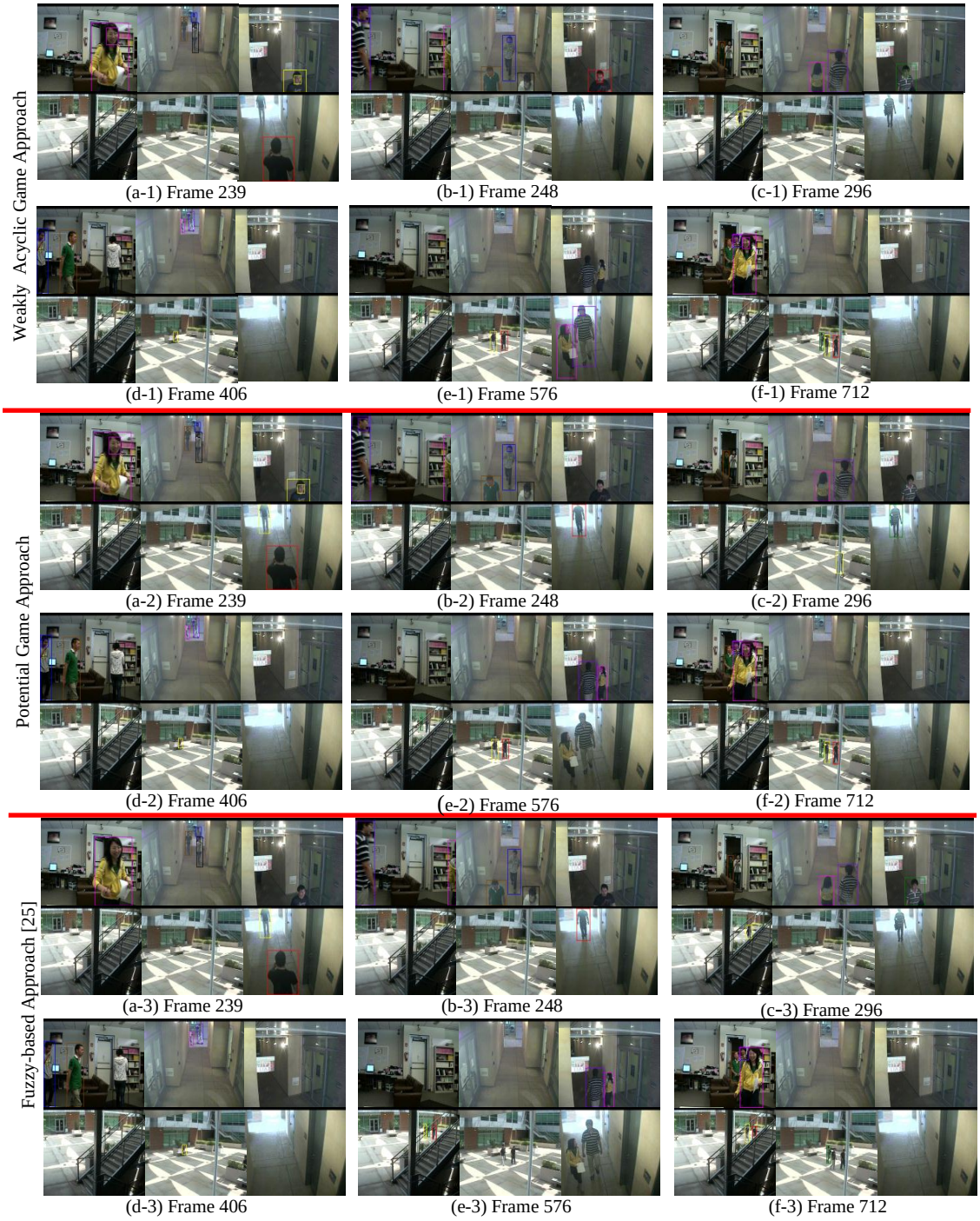


Figure 4.8: Experiment#3. Comparison of example frames by using different approaches in Case 6. The red number is the camera ID. Camera 1 and Camera 5 are used as border cameras. Camera 0, Camera 1, Camera 2 and Camera 5 care for frontal views while Camera 3 and Camera 4 only care for smooth tracking.

C_0	C_1	C_2
C_3	C_4	C_5

take time delay into account when the cameras are awakened by the border cameras in this experiment.

Example frames are shown in Figure 4.8. We can observe that whenever a frontal view is available in an indoor or a corridor camera, the proposed approach selects that camera successfully according to the user's preference. The performance of the proposed approach is consistent as the number of persons goes up. Similarly, the number of handoffs, number of dithering and error rates for the proposed approach are listed in Table 4.7. We have higher error rate in this case because when taking time delay into account, the optimal camera may be missed for 1 or 2 frames when the border camera(s) just awakes it. In that transit frame, the time delay criterion may have a 0 value and lead to a low payoff for this camera such that it is not selected by the system. The error rate caused by this reason with the annotated video is 1.23%.

To link the complexity of the proposed approach to the number of people that can be handled, we show the execution time for a single frame versus different numbers of persons that are handled in Figure 4.9. The overall execution time is roughly linear with the number of persons handled. This is because we do similar calculation for each person

Table 4.7: Comparison of the proposed with other approaches

Case (See Table 4.4	Approach	NH		ND		Error rate							N_F
		AV	PF	AV	PF	AV			PF				
						S_{lost}	S_d	ER_3	S_{lost}	S_d	S_{gfl}	ER_3	
Case 3	WAG	322	341 (9)	9	11 (2)	56	44	4.51%	122 (8)	62 (4)	62 (3)	11.62% (0.54%)	2238
	PG [6]	418	430 (10)	52	70 (3)	186	99	17.11%	312 (10)	199 (4)	269 (7)	25.60% (0.94%)	
	FB [5]	283	290 (7)	7	7 (1)	129	21	26.85%	208 (9)	57 (2)	517 (12)	29.93% (1.03%)	

NH: number of handoffs; ND: number of dithering among frames; AV: Annotated video; PF: Particle filter tracking; WAG: weakly acyclic game; PG: potential game; FB: Fuzzy-based approach; the numbers shown for the results with PF are in the form of average (standard deviation)

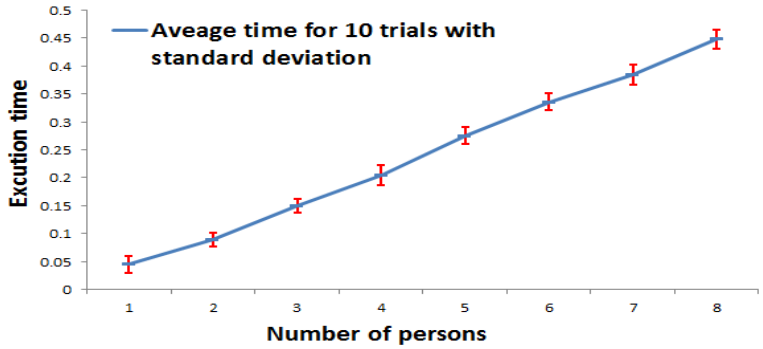


Figure 4.9: Execution time by the proposed approach when altering the numbers of persons handled in Case 6.

in the system and the number of iterations for handling a single person stays almost the same. As the number of persons goes up, there is more information broadcasted by the cameras, but this only accounts for a small part of the computation.

In this experimental case, we compare the proposed approach with both the potential game approach and the fuzzy-based approach. We get similar results by the 3 approaches when a person is visible in only one camera. For example, in Frame 239, Frame 406 and Frame 712, all the 3 approaches select Camera 0 for the persons who are indoor. However, when there are more than one camera available to track a person, the fuzzy-based approach tends to handoff to another camera only when the person is about to leave the FOV of the currently used camera and it does not take view or size of the person into account. Frame 576 is an example of this case. Although the potential game approach does better than the fuzzy-based approach in this sense, it sometimes fails to handoff to the best camera because of the lack of the number of iterations (it takes up to 37 iterations in this case to get the best camera in all frames). It also dithers among cameras a lot since no smoothness is considered by this approach.

The final results are compared in Table 4.7. We can notice that although the fuzzy-based approach does not have the dithering problem, its error rate is still high. This is because it is inherently designed for handing over a camera when an object is on the border of the FOV. Thus, although S_d and S_{lost} can be low in this approach, by using the error definition of Section 4.4.2, this approach raises a very high error rate for the camera where frontal view is requested. Due to the property of the protocol it follows to make the camera hand-off decisions, it is hard to incorporate the view of a person when faced with the hand-off decision among cameras.

4.5.5. Discussion of the Experimental Results

With experiments for six cases, we note the merits of the proposed approach are:

1) The optimal solution for camera selection can be reached with a small number of iterations as compared to the other approaches, [25][26][39]. The experiments are conducted in real-time. We run them on a computer with Intel quad core 3.16 GHz CPU, 4GB memory. Each camera is manipulated as a single thread. Each convergence takes around 0.045 ~ 0.055 second.

2) The system can be task-oriented since different criteria can be applied flexibly.

3) The system, is distributed since no global information is needed during the payoff-based learning process.

The weakness of the proposed approach is that it is sensitive to the tracking errors caused by the tracker. This can be observed from Table 4.5 to Table 4.7 when comparing the results with annotated videos with that by using the particle filter trackers.

4.5. Summary

In this chapter, we model the camera selection and handoff problem as a weakly acyclic game and use the payoff based learning algorithm to get the stable result with guaranteed convergence. We develop the criteria so that the handoffs occur in a smooth manner and take time delay for awakening a camera into consideration. We compare the proposed approach with the potential game approach, both theoretically and experimentally. This comparison shows that the weakly acyclic game approach is much more efficient than the potential game approach. Further, the weakly acyclic game approach removes the requirement of alignment of local and global utilities needed in the potential game approach. So, in the weakly acyclic game approach, the design of criteria and the payoff function for cameras are both more flexible and easier. Since no global information is needed to carry out the camera selection and hand-off in the weakly acyclic game approach, the system is realized in a distributed manner. We show results with real data in 6 different cases, both indoors and outdoors, with different numbers of cameras and persons. We also compared related non-game theoretic approaches [25]. All the results show the efficacy of the proposed approach.

Chapter 5

Coupled Camera Selection and Object Tracking in a Video Network

5.1. Motivations

In the previous two chapters, we introduced two game theoretic approaches to solve the camera selection and handoff problem. This process highly depends on the results returned by the trackers. However, as stated in Chapter 1, there is no single tracker that can perform perfectly in all scenarios. Different types of trackers may achieve different performance under different application scenarios because of their inherent properties. For example, the CamShift tracker [22] is robust for simple scenarios with few occlusions, the particle filter tracker [52] is well suited for occlusions and the series of online boosting trackers [57][58][59] are less sensitive to illumination conditions. There is no single tracker that can tackle with any scenarios. In some circumstances, one tracker fails but another type of tracker may work well. We show this in Figure 5.1.

This motivates us to fuse the performance of all these trackers in a unified manner to achieve a reasonably good tracking result automatically and continuously in different scenarios better than any individual tracker. The difference between the proposed fusion of multiple trackers and the previous approaches is that it is desired that the trackers benefit from the camera selection result. On the other hand, although there are many papers that solve the camera selection problem in different ways (see Chapter 2), but the results of camera selection are not fed back to refine the tracker. This chapter focuses on

performing the camera selection and tracking objects simultaneously so they benefit each other. The approach proposed in this chapter does not have any requirement for online camera calibration.

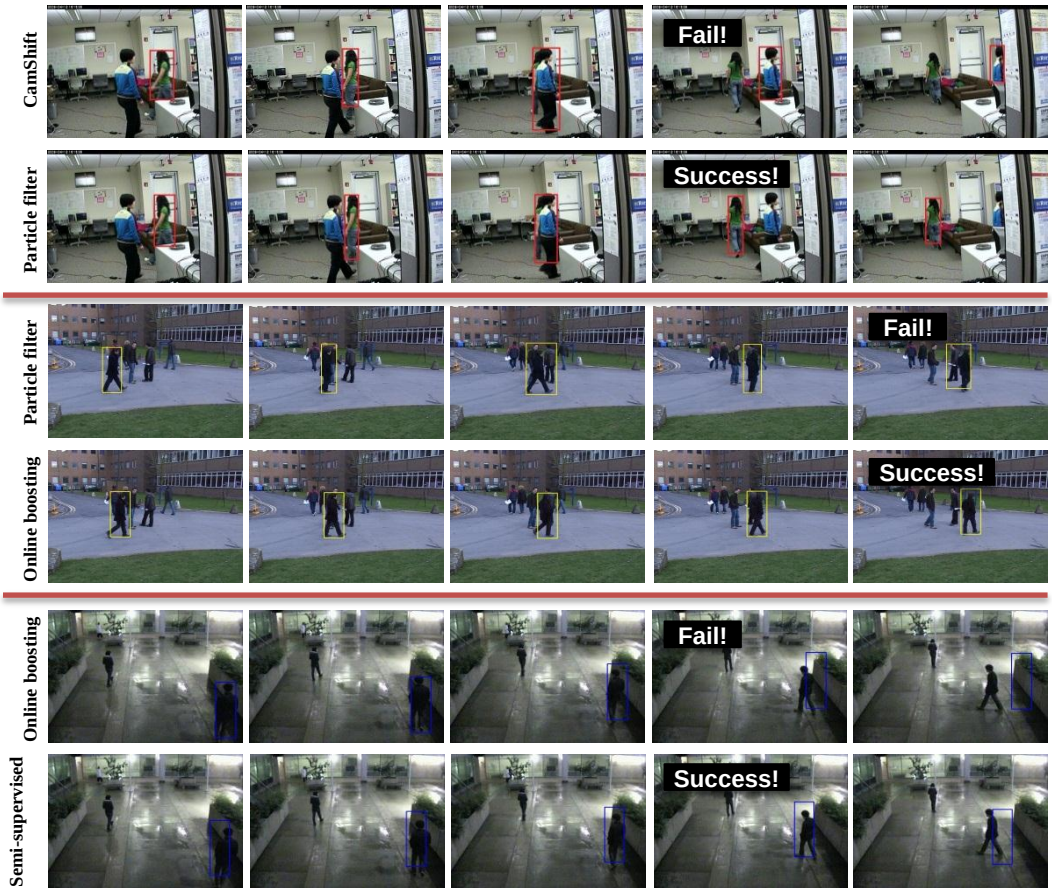


Figure 5.1: No single tracker is good enough for all scenarios. The first two rows show a scenario where the CamShift [18] tracker fails but the particle filter tracker [52] works well. The middle two rows show a scenario where the particle filter tracker [52] fails but the online boosting tracker [57] works well. The bottom two rows show a scenario where the online boosting tracker [57] fails but the semi-supervised online boosting tracker [58] works well. The failure and successful frames are marked as shown in the figure.

5.2. Technical Approach

For the convenience of description in the rest of this chapter, we list the key symbols and their notations in Table 5.1. An overview of the approach is given in Figure 5.2 and

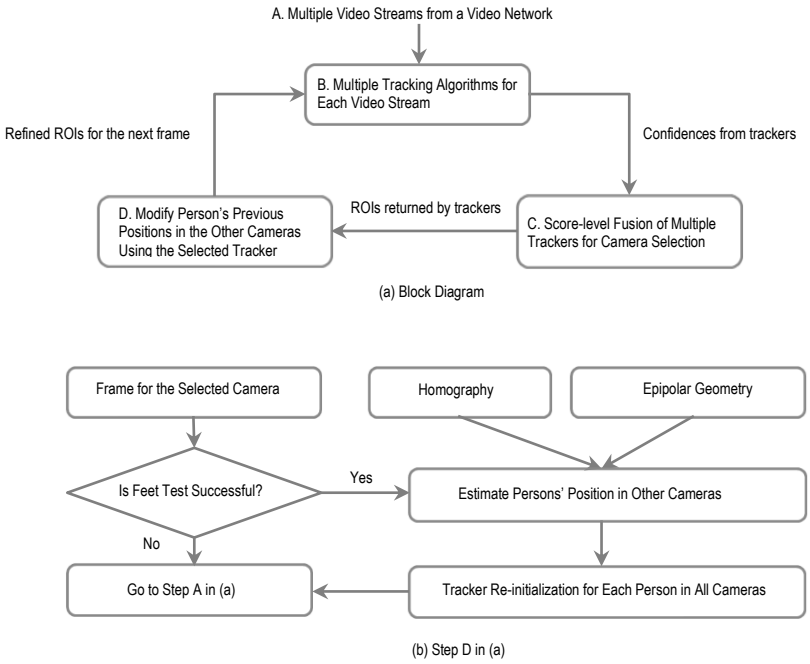


Figure 5.2: Overview of the proposed system.

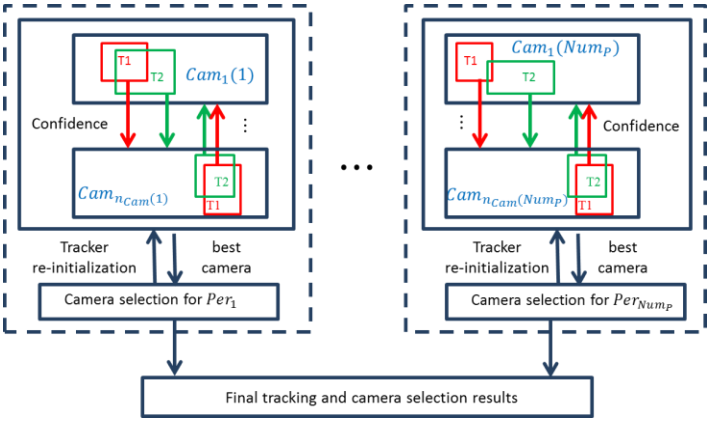


Figure 5.3: Illustration of combining fusion of multiple trackers and camera selection together. In this figure, we assume that there are only two trackers T1 and T2 running in each camera. But in a real application, there are Num_T trackers for each person. The same approach runs for all the persons in the system.

Table 5.1: Symbols and notations used in Chapter 5

Symbols	Notations	Symbols	Notations
Num_P	Number of persons in the system	λ_x	Penalty for selecting tracker *x*
Num_C	Number of cameras in the system	M_i^x	Metric *i*'s value for tracker *x*
Num_T	Number of trackers used for each person	π_i	Weight for metric *i*
Num_M	Number of metrics	γ	Threshold for M_1^x
$n_{Cam}(i)$	Number of cameras that can see person Per_i	X	Object's current centroid position
Per_i	Person *i*	X_c	Center of the image plane
$Cam_j(i)$	j^{th} camera that can see Per_i	X_o	Origin of the image plane
$Score^x$	Final score of tracker *x* for doing camera selection	ω	Parameter needed by the spatial smoothness criterion
$score_{track}^x$	Score to evaluate the tracking quality of tracker *x*	th_1	Threshold of width to height ratio of a bounding box
$score_{camSel}^x$	Score to evaluate the camera selection quality in tracker *x*	th_2	Threshold of occlusion confidence in feet test
$conf^x$	Confidence value of tracker *x*	F_{kj}	Fundamental matrix from Cam_j to Cam_k
t	Current frame number	l_k	Epipolar line in Cam_k
mem	Number of frames that system can remember	P_k	Feet position in Cam_k
α	Parameter deciding the speed of memory fading	H_k	Head position in Cam_k

Figure 5.3. We select Num_T trackers to track persons in a camera network. These trackers run simultaneously in all the cameras. The cameras that can see the same person are allowed to communicate with each to broadcast their trackers' confidences for tracking this person. Then a set of scores, which evaluate both the tracking quality and the camera selection quality, are calculated to select the most appropriate camera for this person. Finally this camera selection result is fed back to re-initialize the trackers and begin tracking for the next frame (This re-initialization is done every *mem* frames. This will be discussed in more details in Section 5.3).

5.2.1. Fusion of Multiple Trackers

We do a score-level fusion of multiple trackers. However, instead of fusing multiple trackers in a single camera, we fuse multiple trackers' information from multiple cameras. Meanwhile, this approach can accommodate to user-provided-criteria for camera selection. When doing the fusion, these criteria for considering a camera hand-off are also taken into account. Another reason why we do a score-level fusion instead of a feature-level one, is that we want to make this approach universal for any combination of multiple trackers. The performance of the fusion will be better than any of the individual tracker. Moreover, whenever new trackers are available, they can be used in the same manner for better results. There are some other possibilities, like a Bayesian framework or the Dempster-Shafer (DST) approach to do the fusion. However, when there is no extensive data to train the system, the performance of these kinds of algorithms are not guaranteed.

Assume that for a particular person Per_i , there are $n_{Cam}(i)$ cameras that can see this person. Suppose we have N_T trackers which run on all cameras. Thus, for each particular Per_i, there are $n_{Cam}(i) \times Num_T$ tracking results all together. We call them hypotheses. For each tracker $x, x\epsilon\{1, \dots, Num_T\}$, we calculate its associated tracking score $score^x_{track}$ and the camera selection score $score^x_{camSel}$. The final score for each tracker, based on which we do the final camera selection, is given below:

$$score^x(t) = score^x_{track}(t) + \lambda_x \cdot score^x_{camSel}(t) \quad (1)$$

$$score^x_{track}(t) = \frac{\alpha \cdot conf^x(t) + (1-\alpha)\sum_{l=t-mem}^{t}(conf^x(l))^\alpha}{\sum_{l=1}^{t} conf^x(l)} \quad (2)$$

$$\lambda_x = \frac{e^{conf^x(t)}}{\sum_{x=1}^{Num_T} e^{conf^x(t)}} \tag{3}$$

$$score^x_{camSel}(t) = \sum_{i=1}^{Num_M} \pi_i M_i^x\,(t) \tag{4}$$

In Equation (1), $score^x(t)$ is calculated for each tracker x. The one with the highest score is selected finally. Since this score is calculated for trackers in all related cameras, when we finally select a tracker for a person, we select a camera simultaneously. So, when calculating this score $score^x(t)$, we consider two aspects:

1) The tracking confidence of the current tracker x, $score^x_{track}(t)$, which has a fading memory of its performance from the previous mem frames up to the current frame. That is, each tracker exponentially discounts the influence of its past tracking quality in the computation of its current tracking quality. The parameter α controls how fast we want the memory to fade away. This formulation allows us to consider the performance of a tracker continuously, such that when we consider to hand off from one camera to another, the temporal smoothness is also taken into account. It also avoids problems from an instantaneous error (such as being distracted by another object) of the tracker, since its performance history will help to rectify the current performance. This term is normalized because we want the two aspects that decide the final score to be balanced. The calculation of $score^x(t)$ will be discussed later in this section.

2) The tracking quality according to the camera selection criteria, $score^x_{camSel}(t)$. When integrating the tracking quality for camera selection, we consider both the current tracker and the other trackers. That is why this score is multiplied by a penalty weight, λ_x. If the current tracker has a lower confidence compared with other trackers, λ will assign a

low weight to it. In this case, even if a tracker may have a high tracking confidence, it may not be selected if it does not meet the camera selection criteria well. For instance, if the size of a tracked person is too small by a tracker, $score_{camSel}^{x}(t)$ for this tracker will be low and the overall score for it will be downgraded. This is the part where we actually fuse the performance of multiple trackers together with camera selection. Traditional camera selection/hand-off approaches are only based on a single tracker. However, in real applications, although the result returned by a tracker may be acceptable for tracking, i.e., it is not too far away from the targeted person, it sometimes is not accurate enough to provide the information needed by the camera selection/hand-off approaches. By applying the proposed idea, the final camera selection result relies more on the information returned by the tracker with higher confidence and, thus, reduces the uncertainty of the camera selection/handoff procedure.

A. Compute the Tracking Confidence

In our experiments, we implement two categories of trackers based on the different features they use: 1) the CamShift tracker (CS) [18] and the particle filter tracker (PF) [52]which use HSI color as the feature; 2) the online boosting tracker (OB) [57], the semi-supervised online boosting tracker (SOB) [58], the multiple instance learning tracker (MIL) [59], which use a feature pool consisting of histogram of orientations, Haar wavelets and the local binary patterns (LBP), and the P-N learning tracker (TLD) [80], which uses ferns [81] as the feature descriptor. The reason why we select these trackers are: 1) They are well known trackers. The implementations of these trackers are publicly available. This will make the evaluation of their performance easy and fair; 2) They can

achieve a real-time performance. Although some other trackers, such as the one in [82], claim for tracking over a long time period, they are either too complicated or need post processing to make online applications unrealizable. For the first category of trackers, we calculate the tracker confidence as the correlation coefficient of the color histogram of the person's bounding box returned by the tracker between the current frame and the previous frame multiplied by the previous frame's confidence. For the second category of trackers, instead of multiplying the previous frame's frame, we multiply the correlation coefficient by the confidence returned by the tracker, which is a weighted summation of a group of weak classifiers, as the tracker confidence (for more information on the calculation, the readers can refer to [57]).

B. Compute the Camera Selection Score

The camera selection score $score_{camSel}^{x}$ is based on the user-supplied metrics. In our experiments, we apply object size $M_1^x(t)$, object's distance to the camera image center $M_2^x(t)$ and the *spatial smoothness* $M_3^x(t)$ [84] as the metrics. The equations of these criteria scores are listed as Equation (5) to Equation (7). In Equation (5), we test the ratio of the object size to the image plane size. Neither too small or too large ratio is desired. γ is a pre-decided threshold for this ratio. The metric score reaches its peak value at this threshold. Smaller or larger than γ downgrades the metric score. In Equation (6), we decide the metric score based on the object's Euclidean distance to the center of the image. This distance is normalized by half the length of the diagonal, which is the longest possible distance. Equation (7) computes the trajectory smoothness based on the object's previous position returned from the previous frame and the current object position. Note

that spatial smoothness refers to the smoothness of the track, whereas the *temporal smoothness* we mentioned previously refers to the usage of a camera to track a person. A camera selection algorithm is temporally smooth means that we do not switch among cameras to track the same person too frequently. This is taken care of in Equation (1). Each criterion score $M_i^x(t)$ is weighted by $\pi_i \in (0,1), i = 1,2,3$. The value of π_i implies the user's bias to this criterion. In this chapter, we get these values empirically and this will be discussed in Section 5.4.

$$M_1^x(t) = \begin{cases} \frac{1}{\gamma} r(t), & when\ r(t) < \gamma \\ \frac{1-r(t)}{1-\gamma}, & when\ r(t) \geq \gamma \end{cases}, r(t) = \frac{\#\ of\ pixels\ inside\ the\ bounding\ box}{\#\ of\ pixels\ in\ the\ image\ plane} \quad (5)$$

$$M_2^x(t) = \frac{|X(t) - X_c(t)|}{|X_o - X_c(t)|} \quad (6)$$

$$M_3^x(t) = \omega \left(\frac{X(t-1)^{\circ} X(t)}{|X(t-1)||X(t)|} \right) + (1-\omega)\left(\frac{2\sqrt{|X(t-1)||X(t)|}}{|X(t-1)| + |X(t)|} \right) \quad (7)$$

5.3. Scene Analysis

Based on the discussion in the previous section, we develop a framework where both camera and tracker selections are integrated. In this section, we will discuss how the trackers can benefit from the camera selection result. As illustrated in Figure 5.2, we feedback the information for the selected camera to rectify the performance of the other trackers. In most of the similar works, a matching among different camera views is done based on the ground plane homography given the fact that a person's feet are always on the ground. The feet are often located near the middle point of the bottom line of the bounding box returned by the tracker. However, this may not be true when the person is

occluded or she/he does not fully appear. So, we do a feet test first such that we can avoid this feedback when there is a low possibility of finding the existence of feet in the bounding box. We first calculate the ratio of the width of the bounding box to the height of the bounding box. In the case that a person does not fully appear, this ratio is usually high. So, if this ratio is higher than a threshold th_1, we say that the person does not fully appear. After this, we locate the feet of a person by using the feet to body ratio according to statistical human proportions (The length of a person's foot is approximately 15% of his or her height [86]. We then test how much of the located feet part is occluded by any other object by comparing the located feet position with the position of bounding boxes of other objects. If ρ% of the feet is occluded, the confidence value returned by the feet test is 1-ρ%. This is to refuse the case when an object is occluded by another object and these two objects are included in the same bounding box. In this case, locating the feet by calculating the feet to body ratio will be affected. We go to the tracker re-initialization step if this confidence value is above a threshold th_2. We show the necessity of the feet test in Figure 5.4. This feet test does not provide a perfect result, but it enhances the performance by approximately 4%.

When the feet test is done, we want to use this feet position in the best tracker to rectify the other trackers in other cameras such that error propagation can be avoided. Unlike most of the other works, in this chapter, we propose the combination of homography and epipolar geometry (see Figure 5.2(b)) for efficient and effective scene analysis.

(b) PETS 2009 frame_0154 view 1_1 (c) PETS 2009 frame_0154 view 1_2

Figure 5.4: Necessity of tracker re-initialization with feet test. In (a) The system incorrectly decides that view 5 is the best camera view to track the occluded person based on the metrics. If there is no feet test, this error will propagate by using the front person's position to re-initialize the tracker for the occluded person in view 1, as shown in (b). However, with a feet test, we know that there is a heavy occlusion and avoid the tracker re-initialization in view 1, as shown in (c). (Different colors are for different trackers.)

The bounding box surrounding a person in a camera's field-of-view (FOV) is determined by the location of the person's head and feet in the image. Under the assumption that there is a single plane (ground plane) that everyone walks on, there exists a homography between any two cameras with this reference plane. Consider that a person is observed in a camera Cam_j. If the feet test is successful, the location of the feet of the person is known in Cam_j. A homography between Cam_j and any other camera (Cam_k) will give the corresponding location of the person's feet in Cam_k. This idea has been used extensively in other works as described in Chapter 2, but it has been restricted only to locating points across views.

We propose using concepts from epipolar geometry (the fundamental matrix) to locate the head of a person. Coupled with homography, this has the added advantage of giving an estimate of the expected pixel-height of the person in other cameras. Continuing the above scenario between cameras Cam_j and Cam_k, if H_j represents the location of the person's head in Cam_j and F_{kj} represents the fundamental matrix from Cam_j to Cam_k, then epipolar line (l_k), in Cam_k given by $l_k = F_{kj}H_j$, is the line in Cam_k

(a) PETS 2009 frame_0154 view

(b) PETS 2009 frame_0154 view 1_1

(c) PETS 2009 frame_0154 view 1_2

Figure 5.5: Necessity of tracker re-initialization with homography only and with epipolar geometry. (a) Selected view using online boosting. (b) Particle filter tracker re-initialization without providing the epipolar line. The re-initialized tracker in (b) still includes other person in it, which is a mistake. (c) The re-initialized tracker by using homography and epipolar geometry together avoids the problem in (b). Only a single person is in the bounding box.

passing through the epipole and contains the location of the persons head, H_k. Enforcing the condition that all objects are always upright, H_k is the point on l_k with the same *x* coordinate as P_k, where $P_k = (P_k^x, P_k^y)$ is the feet position matched from homography between Cam_j and Cam_k, i.e., $H_k = (H_k^x, H_k^y) = \{(H_k^x, H_k^y) \in l_k : H_k^x = P_k^x\}$.

This idea is illustrated in Figure 5.5. Note the change of the heights of the re-initialized tracker between (b) and (c). The frequency of this re-initialization is related to the length of a tracker's memory. As described in Equation (2), there are *mem* frames accounted for calculating tracking score at the current frame t, $score_{track}^{x}(t)$. To make the usage of the tracking performance history meaningful, the re-initialization must take place every mem' frames, where $mem' \geq mem$. In our experiments, we make $mem' = mem$. Thus, the re-initialization of trackers is performed every *mem* frames.

An example frame for the tracker re-initialization process is shown in Figure 5.6.

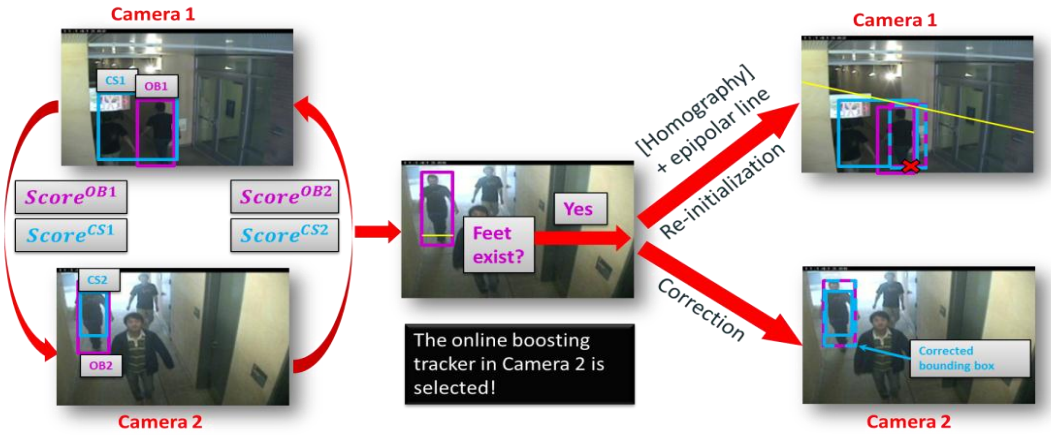


Figure 5.6: Process for tracker re-initialization. The left part of the figure shows the fusion process of the CamShift tracker and the online boosting tracker. The online boosting tracker in Camera 2 is finally selected after calculating the score for each tracker. A feet test is then executed. The success of the feet test triggers the process of tracker re-initialization. The information of the selected online boosting tracker in Camera 2 is used to correct the location of the other tracker (the CamShift tracker) in the same camera (Camera 2). Homography and the epiplolar line for the other camera, Camera 1, is computed to re-initialize the two trackers in Camera 1.

5.4. Experimental Results

5.4.1. Datasets

To show the robustness of the proposed approach, we test the proposed approach in several widely used publicly available datasets listed in Table 5.2.

The reasons that we choose these datasets are: 1) These datasets contains many challenging scenarios, such as different illumination conditions, different sizes of objects, different extents of occlusions, different lengths of videos, etc. 2) The homography of some of the datasets, such as the cvlab datasets [87], are provided together with the datasets. For the other dataset it is easy for us to calculate the homography and fundamental matrices between different camera views. We calculate these matrices by

Table 5.2: Datasets used for experiments

Experimental cases	Datasets	Environment	Num_P	Num_C	Length
Case 1	cvlab Laboratory sequence1	indoor	4	4	5000
Case 2	cvlab Labortory sequence2	indoor	6	4	2951
Case 3	PETS 2009 S2.L1	outdoor	9	5	795

selecting corresponding points manually from some of the frames and then testing them on others. We use the views with the error in the range of 2~10 pixels.

A. Parameters and Initialization

For the parameters used in this chapter, we run the algorithm for the first 125 frames of each dataset using different parameter values in the range of (0,1) with step 0.05 (*mem* is tested in the range of (0, 20) with step 1) and compare the results with the manually annotated ground truth data using an online available tool, ViPER-GT [85]. The best performed parameters are listed in Table 5.3. Since the confidence value of different trackers are based on different principles, we also do a range normalization of these confidence values within these 125 frames when comparing these confidences. We initialize the positions and IDs of people who appear in these 125 frames manually to provide a reliable start. Whoever is marked in the first 125 frames will be considered in the rest of the sequence. Those who enter the scene after the 125^{th} frame will be ignored. An automatic way could be running human detection algorithm at each frame and doing data association for every detected person. But since we want a real-time application based on an accurate initialization and current real-time human detection algorithms are not 100% accurate, we do a manual initialization. Thus, our results can be evaluated without the consideration for the performance of human detection algorithms.

Table 5.3: Parameter values in each dataset

Experimental cases	α	π_1	π_2	π_3	mem	th_1	th_2
Case 1	0.7	0.4	0.3	0.3	5	0.3	0.4
Case 2	0.7	0.5	0.3	0.2	5	0.3	0.4
Case 3	0.5	0.55	0.25	0.2	10	0.5	0.5

B. Ground-Truth and Error Metrics

In our experiments, since we do tracking and camera selection simultaneously, two kinds of ground-truth are needed: the tracking ground-truth and the camera selection ground-truth. 1) Tracking ground-truth. We use ViPER-GT [85] to annotate the objects' position in the video sequences manually. The obtained positions are used as the ground-truth for tracking. 2) Camera selection ground-truth. Based on the tracking ground-truth and assuming the confidence of the manually labeled tracker is always 1, we calculate the camera selection metric scores $score_{camsel}^{x}$ using the object's ground-truth position and select the camera with the highest score. This is used as the camera selection ground-truth. In the following experiments, we compare the experimental results with the ground-truth data. If the selected camera is the same with the camera selection ground-truth, and the bounding box returned by the selected tracker is η (70%-150% in our experiments as shown in Figure 5.7) overlapped with the tracking ground-truth, we treat it as a correct frame, otherwise, it is treated as an error. The error rate in each case is defined as:

$$E_i = \frac{Number\ of\ error\ frames\ for\ case\ i}{\sum_{t=1}^{Num_F} n_P(t)},$$

where Num_F is the number of frames, $n_P(t)$ is the number of persons in the current frame *t*.

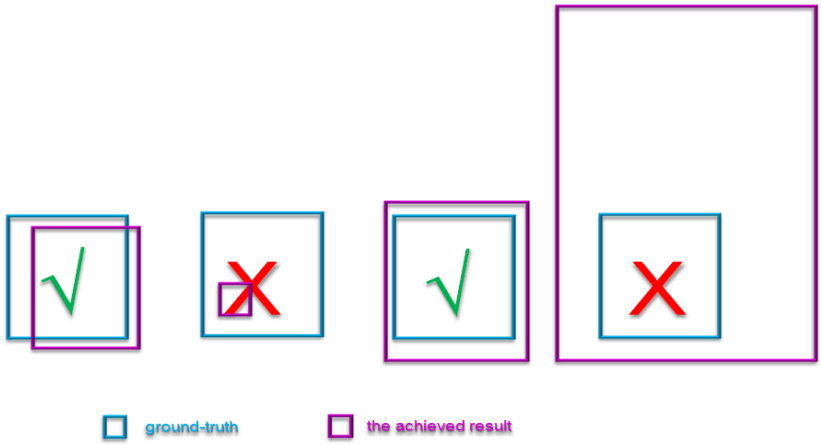


Figure 5.7: Illustration for error definition.

5.4.2. Experiments

In this section we show various experimental results and comparisons with different datasets.

A. Individual Trackers vs. Fusion of Multiple Trackers in a Single Camera

In this section, we will show the effectiveness the effectiveness of fusion of multiple trackers first. The advantages of integrating camera selection into the tracking process will be shown in the next section. Hence, in this section, we only show tracking in a single camera. Results for the next section will be multi-camera-based.

In Figure 5.7, we show results from the PETS 2009 dataset. As can be seen, any single tracker cannot handle the tracking task for a long time reliably. There are always some scenarios that a tracker may partially or fully lose the object. In some cases, the tracker may not be able to recover from a tracking error and let the error propagate. However, if we apply multiple trackers simultaneously, because of the different inherent

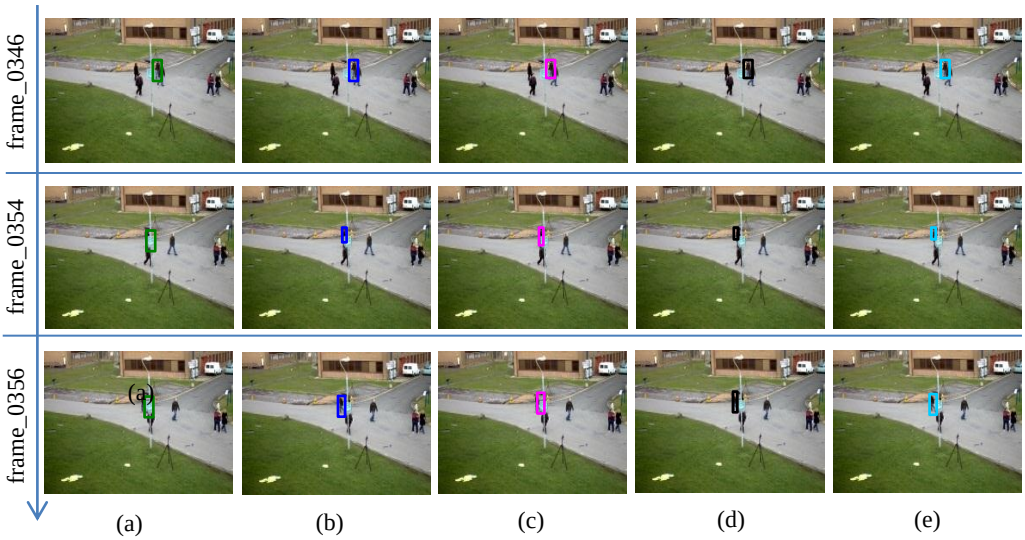


Figure 5.8: Effectiveness of fusion of multiple trackers (with no tracker re-initialization). PETS 2009 frame 346-356, view 1. (a) Use the CamShift tracker only. (b) Use the particle filter tracker only. (c) Use the Boosting tracker only. (d) Use the semi-supervised boosting tracker only. (e) Fusion of the above trackers. The fusion approach considers the criteria and selects the tracker with the best score. As a result, the semi-supervised boosting tracker is selected in frame_0346 and frame_0354, the particle filter tracker is selected in frame_0356.

principles of the tracker, when one of the trackers performs badly, some of the others may be operating properly, such that we always choose the best performing tracker and use this to rectify the others. For example, in Figure 5.7 (a), when the person is occluded by the pole, the tracker loses this person and stays there forever. In Figure 5.7 (b), although the tracker follows the person in each frame shown, it partially loses the person sometimes, i.e., the confidence of the tracker is sometimes low. Similarly, in Figure 5.7 (c) and (d), the boosting and semi-supervised boosting trackers' confidences cannot keep a high value in every frame. The fusion scores of the above trackers from frame_0346 to frame_0356 are shown in Figure 5.8. In contrast, when we apply the fusion approach described in Section 5.2, we get the trajectory most close to the ground truth, as shown in Figure 5.9. Note that the trajectories we show are the 2D camera image coordinates in

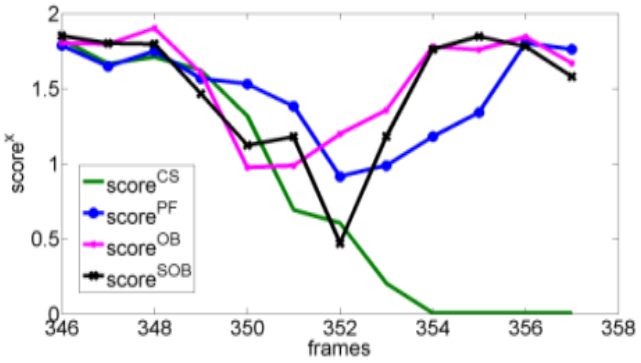


Figure 5.9: Scores of each individual trackers.

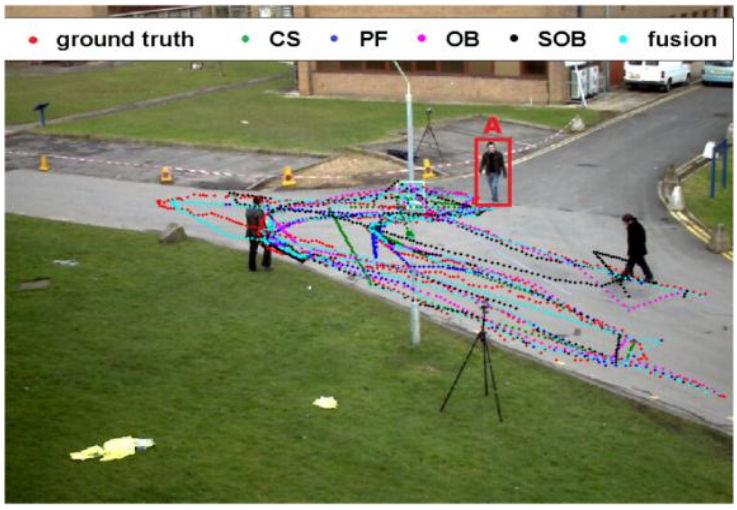


Figure 5.10: Comparison of individual trackers' tracks with the track obtained by the fusion of different trackers.

view 1 of the PETS 2009 S2. L1 dataset. Some of the trajectories are shorter than the ground truth because the trackers either stop due to low confidence (the boosting and semi-supervised boosting trackers) or stays at a wrong location and cannot recover from there. This problem is solved by introducing the tracker re-initialization mechanism.

Similarly, we show some example frames from experimental Case 1 in Figure 5.10. Quantitative results are shown in Table 5.4. Note that results in Table 5.4 are for the persons shown in Figure 5.7 and Figure 5.10 only. A full result for each person in each case will be presented in next subsection. Since the tracking results can be improved by

Table 5.4: Comparison of error rates by using individual trackers and fusion of multiple trackers. Results are shown as mean (standard deviation)

CS: CamShift tracker; PF: particle filter tracker; OB: online boosting tracker; SOB: semi-supervised online boosting tracker; Fusion: fusion of these 4 trackers

Experimental Cases	CS	PF	OB	SOB	Fusion
Case 1	32.9% (2.38%)	25.2% (2.12%)	21.8% (1.98%)	20.9% (1.76%)	19.8% (1.21%)
Case 3	33.6% (2.965)	28.3% (2.54%)	23.4% (1.86%)	23.6% (2.01%)	16.6% (2.09%)

fusion of multiple trackers, in the following experiments, we will always use multiple trackers and fuse their results. However, due to the real-time constraint, we cannot apply a large number of trackers. Besides, if some trackers work worse than other trackers for most the time, then, there is no need to apply them. In Table 5.5, we compare some combinations of different trackers tested on the most complicated case, the PETS 2009 dataset (the first 125 frames' results). The factors considered when combining trackers are the process speed and the obtained performance. Plus, to increase the robustness in different scenarios, we do not want all the trackers use the same set of features. The trackers compared and the reasons why these trackers are selected were discussed previously in Section 5.2.1 A. Since this comparison is done to decide which trackers to be selected for fusion, no camera selection result is fed back in this comparison. From this table, we can conclude that the combination of the CamShift tracker, particle filter tracker, online boosting tracker and semi-supervised online boosting tracker gives a good trade-off between speed and performance. This is the combination we have for fusion for results shown in Table 5.4. We will also use this combination of trackers in the following experiments.

B. Integrating Camera Selection and Tracking in Multiple Cameras

In the following, we show the effectiveness of the proposed tracker re-initialization scheme using the information fed back by the camera selection result. The datasets used are shown in Table 5.2.

The cvlab datasets [87] are both indoor with changing illumination and occlusion. The object size is relatively large in these two datasets, so it is easier for the trackers to track objects in these datasets. The ground plane homography matrices between different camera views are provided. We show Person 1's (in Sequence2) camera selection results with tracker re-initialization with camera selection feedback in Figure 5.11 (a). From the figure we can see that although some trackers are not selected, they can still be selected in the future frames after re-initialization. For example, in frame_0544, the online

Table 5.5: Comparison of different combinations of trackers. The error rates are shown as mean (standard deviation)

CS: CamShift tracker; PF: particle filter tracker; OB: online boosting tracker; SOB: semi-supervised online boosting tracker; MIL: multiple instance learning tracker; TLD: P-N learning tracker

Combination of trackers	Process speed (fps)	Error rates
CS/PF/OB/TLD	17	14% (1.27%)
CS/PF/OB/MIL	12	16% (1.98%)
CS/PF/OB/SOB	**15**	**12% (0.96%)**
CS/PF/SOB/MIL	13	12% (1.34%)
CS/PF/SOB/TLD	14	11% (2.78%)
CS/PF/MIL/TLD	12	13% (2.55%)
CS/OB/SOB/MIL	10	23% (2.18%)
CS/OB/SOB/TLD	10	21% (2.00%)
CS/OB/MIL/TLD	11	20% (1.58%)
CS/SOB/MIL/TLD	11	22% (2.17%)
PF/OB/SOB/MIL	8	13% (1.22%)
PF/OB/SOB/TLD	10	10% (0.39%)
PF/OB/MIL/TLD	9	12% (1.11%)
PF/SOB/MIL/TLD	7	11% (0.66%)
OB/SOB/MIL/TLD	5	19% (1.11%)

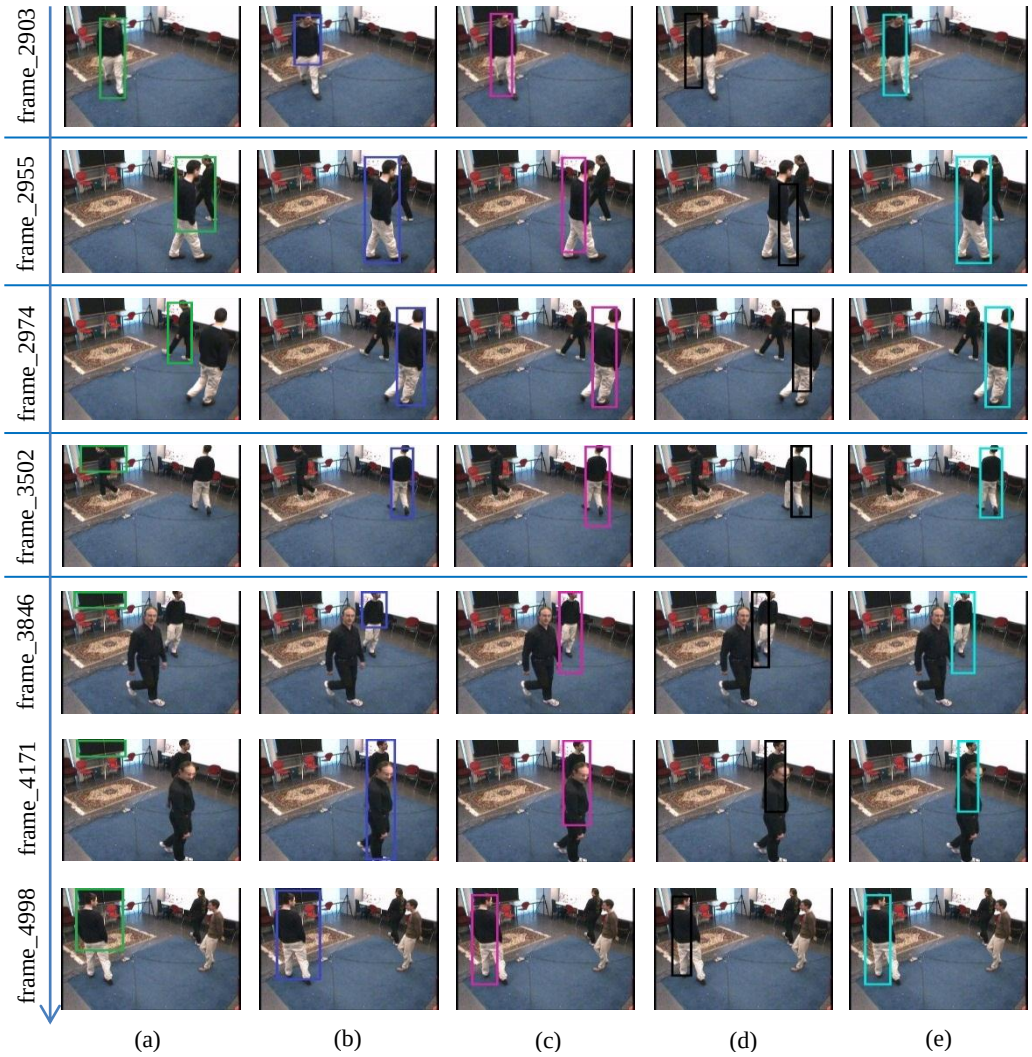
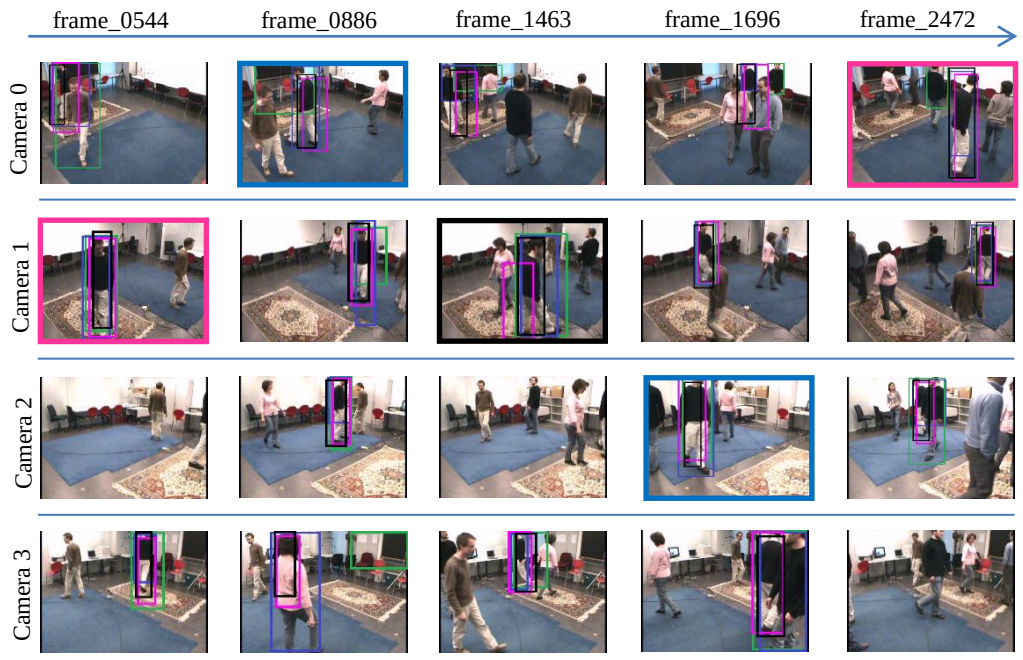
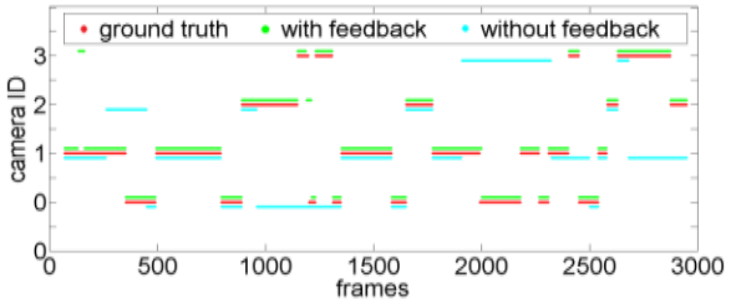


Figure 5.11: Individual trackers vs. fusion of multiple trackers. Case 1. cvlab Laboratory sequence1, example frames from frame 2903-4998. Column (a): using the CamShift tracker [18] only. Column (b): using the particle filter tracker [52] only. Column (c): using the online boosting tracker [57] only. Column (d): using the semi-supervised online boosting tracker [58] only. Column (e): fusion of the above 4 trackers. From frame_2955 to frame_2974, the Camshift tracker is easily distracted by other similar object. From frame_3502, the CamShift tracker stays there until frame_4998. Although it seems that the CamShift tracker is recovered from the error, however, this is an occasional case, where the target object happens to be in similar color (black) with the object distracted the tracker. Particle filter works fine for most of the frames in this simple case. But it cannot distinguish different objects well when they are partially occluded, as shown in frame_4171. The online boosting tracker and the semi-supervised online boosting tracker can follow the object in all the shown frames, but they do not always provide a precise bounding box. Fusion of these 4 trackers selects the best one according to their tracking scores and provides the most desirable results.



(a)



(b)

Figure 5.12: Comparisons of using multiple trackers with and without performing tracker re-initialization with camera selection feedback. (a) Example frames in Case 2. (b) Overall camera selection results for Person 1 with and without camera selection feedback. It is visually obvious that the result with camera selection feedback has less deviation from the ground-truth.

boosting tracker is selected for tracking Person1, but in frame_0886, the particle filter tracker is selected. Similarly, in frame_0886, the CamShift tracker in Camera 3 is distracted by the blackboard, it recovers after re-initialization and provides reasonable results in frame_1463 (although it is not selected). Note that the selection process not

only considers the tracking accuracy, but also the camera selection metrics. For example, in frame_1696, trackers in Camera 1 provide good tracking results, but the system selects the particle filter in Camera 2. This is because the $score^x$ in Camera 1's view are downgraded by the $core^x_{camSel}$. Both the size and position metrics have a higher score in Camera 2. This explains why the particle filter tracker in Camera 2 is selected in frame_1696 and the online boosting tracker in Camera 0 is selected in frame_2472. Figure 5.11 (b) shows the overall camera selection results for Person 1. In this figure, both results with and without feeding back the camera selection information to do tracker re-initialization are shown as a comparison. The results with feedbacks comply with the ground-truth much better than that without feedbacks.

In Figure 5.12, we show results for using the PETS 2009 dataset. There are originally 7 views provided in the dataset, but we use only 5 of them (View 3 and view 4 are not included.) This is because we calculate the homography and fundamental matrices for this dataset by manually picking corresponding point pairs. We do this on some of the frames and test the calculated matrices on the others. The homography matrices and their error mean and standard deviation are shown in Table 5.6. The errors of all the calculated fundamental matrices belong to the defined error range. However, the errors of some of the homography matrices are large, so we exclude those cameras which cause these errors, or do not apply the tracker re-initialization step between the camera views with large errors. The reason for large errors of the homography matrices are: 1) the ground plane is not pure flat. 2) Since we manually pick points, in some views, there are not enough good

Table 5.6: Homography matrices and their errors

√: valid; NA: not applicable; $\boldsymbol{\mu}$: error mean; $\boldsymbol{\sigma}$: error standard deviation

view	**1**			**5**			**6**			**7**			**8**		
		μ	σ		μ	σ		μ	σ		μ	σ		μ	σ
1				√	10	8	√	4	3	√	7	3	√	6	3
5	√	10	8				NA	14	1	√	6	3	√	8	5
6	√	4	3	NA	14	1				√	8	4	√	4	3
7	√	7	3	√	6	3	√	8	4				√	10	9
8	√	6	3	√	8	5	√	4	3	√	10	9			

corresponding point pairs on the ground plane. The trackers are reinitialized every 8 frames.

We can observe that because of the tracker re-initialization, some of the abandoned tracker can be used in future frames. For example, in Figure 5.9 frame_0318 view 1, the particle filter tracker for person 1 is distracted by the other person passing by. The online boosting tracker in view 8 is selected for person 1 and its information is used to re-initialize all the other trackers at frame_0323, such that in the succeeding frame_0324, the particle filter tracker for this person is recovered. Similar case is the resumption of the online boosting tracker for person 2. In frame_0322 view 5, the online boosting tracker returns a very low confidence such that it loses the person, and the semi-supervised boosting tracker in view 7 is selected for person 2 in frame_0322. After using the information from the semi-supervised boosting tracker to re-initialize the other trackers at frame_0323, the online boosting tracker in view 5 is resumed. As a comparison, we show the track for person 1 in view 1 (the longest view in the video) with and without tracker re-initialization in Figure 5.13.

The necessity of each of the proposed steps can is evident by comparing the experiments in the previous subsection, where only fusion of multiple trackers are

Figure 5.13: Effectiveness of the tracker re-initialization scheme. PETS 2009 frame_0314-frame_0324. The cameras selected for person 1 are boxed with solid line, the cameras selected for person 2 are boxed with dashed line. The box color stands for the tracker selected. green: the CamShift tracker; blue: the particle filter tracker; magenta: the online boosting tracker; black: the semi-supervised boosting tracker.

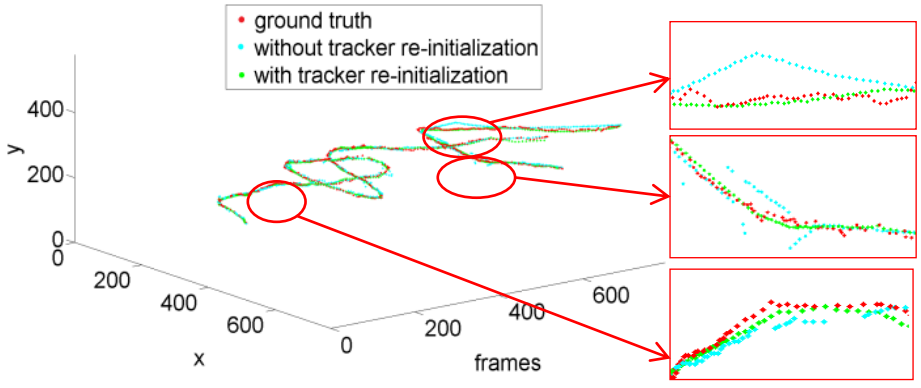


Figure 5.14: Tracker re-initialization improves tracks. For a clearer observation, we give a close-up view of some parts of the trajectory on the right. As can be seen, the green trajectory with tracker re-initialization is much closer to the ground truth.

Table 5.7: Error rates in different cases. Results are shown as mean (deviation).

E1: without tracker re-initialization; E2: without feet test (but with tracker re-initialization with epipolar geometry); E3: without epipolar geometry (but with feet test); E4: the proposed approach (with tracker re-initialization, feet test and epipolar geometry)

Table 5.7-1 Case 1

	E1	E2	E3	E4
Person 1	19.8% (0.33%)	11.4% (0.48%)	9.9% (0.86%)	7.2% (0.39%)
Person 2	15.3% (0.96%)	10.2% (1.58%)	9.8% (0.77%)	5.2% (1.57%)
Person 3	16.9% (1.01%)	7.7% (0.66%)	7.2% (0.91%)	4.5% (0.79%)
Person 4	19.9% (2.33%)	11.6% (1.11%)	10.2% (1.09%)	6.4% (1.22%)

Table 5.7-2 Case 2

	E1	E2	E3	E4
Person 1	17.3% (1.92%)	10.3% (0.93%)	8.9% (2.01%)	6.2% (1.18%)
Person 2	13.2% (1.09%)	9.2% (1.38%)	7.8% (1.96%)	4.6% (0.73%)
Person 3	14.4% (0.94%)	7.7% (0.33%)	6.2% (0.88%)	3.5% (0.87%)
Person 4	16.9% (1.10%)	10.7% (2.07%)	9.2% (1.35%)	6.5% (0.74%)
Person 5	9.8% (1.01%)	8.4% (0.975)	5.6% (0.995)	3.2% (1.44%)
Person 6	11.7% (1.59%)	9.2% (1.87%)	6.8% (1.23%)	4.6% (1.56%)

Table 5.7-3 Case 3

	E1	E2	E3	E4
Person 1	16.6% (0.69%)	12.7% (1.05%)	8.6% (1.55%)	7.2% (1.72%)
Person 2	19.3% (0.89%)	11.1% (0.44%)	9.9% (0.55%)	7.6% (0.46%)
Person 3	22.6% (1.66%)	11.7% (1.29%)	11.2% (0.88%)	8.9% (1.33%)
Person 4	11.7% (1.76%)	6.9% (0.83%)	5.9% (1.49%)	4.9% (1.09%)
Person 5	13.6% (1.33%)	9.8% (1.98%)	8.9% (2.19%)	6.6% (2.01%)
Person 6	15.8% (1.66%)	10.3% (2.22%)	9.8% (1.98%)	6.9% (1.57%)
Person 7	16.1% (0.93%)	12.2% (0.99%)	9.6% (1.59%)	7.1% (1.73%)
Person 8	10.9% (1.88%)	7.4% (2.21%)	8.2% (1.26%)	3.2% (2.97%)
Person 9	15.9% (1.77%)	9.7% (1.67%)	7.7% (1.43%)	5.7% (1.75%)

performed, and those in this subsection. The quantitative results for each person in each case are shown in Table 5.7, from which the necessity of each proposed step, i.e., the feedback of camera selection result to do tracker re-initialization, the feet test and introducing the epipolar geometry for data association, is evident. By comparing the results, we can conclude that feeding back the camera selection results contributes to the overall results most, which improves it by around 10% at average, while introducing the

feet test and the epipolar geometry improve the final results by around 4% and 3% respectively, at averages.

C. Camera Selection without Fusion of Multiple Trackers

In Section 5.4.2 A, we compared individual tracker versus fusion of multiple trackers without camera selection. In Section 5.4.2 B, we compared fusion of multiple trackers with and without tracker re-initialization by feeding back the camera selection results. In this section, we compare doing camera selection by using individual trackers and fusion of multiple trackers. Different from Section 5.4.2 A, this will show the impact of fusion of multiple trackers on the camera selection approaches. To show the necessity of fusing multiple trackers, we do the comparison based on two different camera selection frameworks: 1) use the fuzzy-based approach in [25]; 2) use the same camera selection approach as in the previous experiments. To make the comparison fair, we do two comparisons:

1) With and without fusion of multiple trackers by the fuzzy-based approach, but with different ground-truth. This comparison is shown in Table 5.8-1 (with the ground-truth described in Section 5.4.2 B) and Table 5.8-2 (with the ground-truth recalculated using the fuzzy-based rules in [25]).

2) With and without fusion of multiple trackers by performing camera selections based on calculating the camera selection score according to our criteria, i.e. the same camera selection approach as the experiments in the previous subsection. The ground-truth data are kept the same as those descripted in Section 5.4.2 B. This comparison is shown in Table 5.9.

Table 5.8: Results by using the rules in fuzzy-based approach. Results are shown as mean (standard deviation)

CS: CamShift tracker; PF: particle filter tracker; OB: online boosting tracker; SOB: semi-supervised online boosting tracker; fusion: fusion of the above 4 trackers.

Table 5.8-1 Error rates based on the camera selection ground-truth in Section 5.4.1 B

	CS	PF	OB	SOB	Fusion
Person 1	19.9% (1.08%)	15.3% (1.43%)	11.8% (1.32%)	12.2% (2.12%)	8.9% (2.11%)
Person 2	21.9% (1.97%)	14.2% (1.55%)	11.2% (1.46%)	11.7% (1.71%)	9.6% (1.92%)
Person 3	22.6% (2.11%)	17.7% (1.67%)	12.3% (2.44%)	12.6% (1.79%)	8.7% (2.31%)
Person 4	23.9% (1.89%)	18.7% (2.22%)	16.9% (1.19%)	12.4% (1.56%)	10.7% (1.74%)
Person 5	18.8% (1.92%)	15.4% (1.98%)	11.7% (1.22%)	10.9% (1.34%)	9.2% (2.02%)
Person 6	16.9% (1.895)	13.2% (0.78%)	9.9% (1.66%)	10.1% (0.54%)	5.2% (1.38%)
Person 7	20.7% (0.33%)	15.3% (1.92%)	13.8% (0.66%)	11.1% (0.48%)	9.6% (0.88%)
Person 8	29.8% (0.54%)	17.1% (0.99%)	13.1% (1.01%)	12.8% (1.28%)	9.8% (0.44%)
Person 9	18.3% (1.11%)	17.6% (0.48%)	12.2% (1.09%)	10.5% (1.32%)	7.7% (2.11%)

Table 5.8-2 Error rates based on the fuzzy-based rule ground-truth

	CS	PF	OB	SOB	Fusion
Person 1	32.9% (2.44%)	32.1% (2.98%)	31.1% (1.38%)	29.8% (1.96%)	26.2% (2.18%)
Person 2	34.6% (2.87%)	37.9% (3.10%)	32.9% (2.34%)	30.1% (3.03%)	27.6% (2.68%)
Person 3	29.9% (2.76%)	29.6% (1.26%)	26.7% (2.22%)	24.4% (2.11%)	19.9% (2.21%)
Person 4	29.2% (2.98%)	32.3% (2.31%)	22.3% (2.33%)	23.7% (2.13%)	16.4% (2.07%)
Person 5	31.1% (2.78%)	29.9% (1.99%)	28.4% (3.04%)	21.0% (2.98%)	25.6% (2.19%)
Person 6	38.7% (3.01%)	38.1% (2.45%)	36.2% (2.99%)	33.3% (2.97%)	29.1% (2.11%)
Person 7	29.9% (2.99%)	25.6% (2.53%)	26.0% (1.89%)	23.2% (1.77%)	17.7% (1.57%)
Person 8	34.5% (2.56%)	35.1% (2.88%)	29.8% (3.02%)	26.9% (1.69%)	20.3% (1.73%)
Person 9	29.2% (2.99%)	28.8% (2.67%)	22.3% (1.99%)	24.5% (2.56%)	18.5% (1.59%)

Table 5.9: Results by using the criteria proposed in Section 5.2.1 B (shown as mean (standard deviation)

CS: CamShift tracker; PF: particle filter tracker; OB: online boosting tracker; SOB: semi-supervised online boosting tracker; fusion: fusion of the above 4 trackers.

	CS	PF	OB	SOB	Fusion
Person 1	18.8% (1.22%)	16.4% (0.79%)	14.9% (0.94%)	13.9% (0.82%)	7.2% (0.65%)
Person 2	19.9% (1.36%)	14.1% (1.44%)	18.1% (0.87%)	14.2% (0.79%)	7.6% (0.48%)
Person 3	18.9% (1.87%)	15.7% (1.52%)	13.9% (1.11%)	11.1% (0.48%)	8.9% (0.38%)
Person 4	17.9% (1.32%)	17.7% (1.28%)	12.2% (1.02%)	8.9% (0.82%)	4.9% (0.45%)
Person 5	18.2% (2.43%)	14.3% (1.97%)	10.7% (0.66%)	9.4% (0.83%)	6.6% (0.87%)
Person 6	20.8% (2.18%)	14.2% (1.88%)	11.8% (0.69%)	12.3% (0.92%)	6.9% (0.48%)
Person 7	19.3% (1.44%)	16.4% (1.79%)	14.2% (0.77%)	13.2% (0.69%)	7.1% (0.33%)
Person 8	16.5% (1.89%)	13.1% (1.54%)	9.8% (0.89%)	6.9% (0.73%)	3.2% (0.55%)
Person 9	17.7% (1.77%)	15.6% (1.83)	18.6% (0.94%)	10.1% (1.01%)	5.7% (0.77%)

In both cases, we feed back the camera selection results; no matter the system uses an individual tracker or fusion of multiple trackers. The comparisons are done on the PETS 2009 dataset. Results for the first comparison are shown Table 5.8. Results for the second comparison are shown in Table 5.9. Table 5.8-1 and Table 5.8-2 show different results because different criteria are applied when evaluating the results. If we only consider camera handoffs taken on the border of camera FOVs, i.e. according to the fuzzy-based rules, then, Table 5.8-1 applies. If we consider several criteria discussed in Section 5.2.1 B, then, Table 5.8-2 applies. In both cases, we can see the improvements obtained by the fusion of multiple trackers. By comparing Table 5.9 and Table 5.4, we can see the contribution of integrating camera selection into the system improves the single trackers' performance as well, because the trackers are re-initialized from time to time. Comparing Table 5.8-2 and Table 5.9, we observe that if multiple user-supplied criteria are used for camera selection, then the proposed framework works better than the fuzzy-based approach.

5.5. Summary

In this chapter, we proposed an approach to do camera selection using a score-level fusion of multiple state-of-the-art trackers with a novel tracker re-initialization scheme. The fusion process does not only take into account a tracker's accuracy, but also the tracked object's attributes according to camera selection criteria. A tracker with a higher tracking performance but worse camera selection attributes may be avoided in the final selection. In this way, the fusion of multiple trackers and the camera selection are done at the same time. To benefit from the combination of fusion of multiple trackers and camera

selection, the selected tracker's information is used to re-initialize the locations of other trackers. This is achieved by making use of the pre-calculated homography and epipolar geometry together. A feet test is performed before this to make the re-initialization more reliable.

The proposed approach is evaluated using several widely used public datasets. The effectiveness of the proposed approach is shown by providing many comparisons. We show the necessity of applying multiple trackers than using a single tracker only and the necessity of doing scene analysis with feet existence test and the epipolar geometry. The proposed joint optimization approach provides robust tracking and camera selection: 1) The closed-loop framework with feedback from the camera selection results to re-initialize the trackers improves the overall performance by 9.88% on the average. 2) The feet test before camera selection feedback improves the overall performance by 3.91% on the average; 3) The usage of epipolar geometry improves the overall performance by 3.02% on the average; 4) Fusion of multiple trackers, comparing with individual trackers, improves the overall results by 10.1% on the average.

Chapter 6

Auction-based Dynamic Camera Grouping with Active Control

After introducing the camera selection and handoff problem, in this chapter, we will look into the camera active control.

6.1. Problem Formulation and Notations

A block diagram in Figure 6.1 provides an overview of the proposed approach, which is described in this section. In our experimental environment some cameras overlap in their field-of-views (FOVs) while some others do not overlap. The virtual auctioneer announces the location of persons in the system and the cameras send out their bids. Most

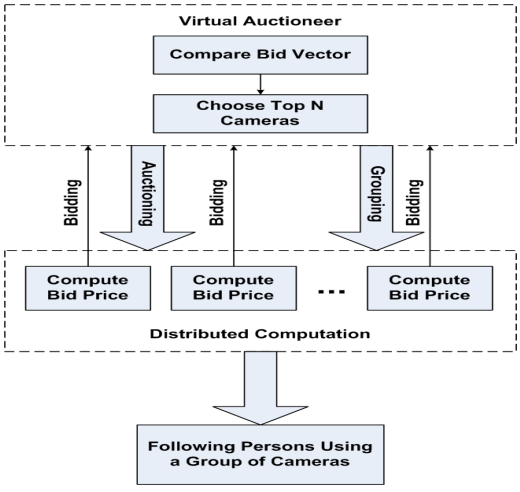


Figure 6.1: Overview of the auction-based approach. A virtual auctioneer holds an auction for a person P_i. All the available or potentially available cameras compute their bid prices for P_i locally and submit this information to the auctioneer. The cameras are grouped automatically for P_i based on their bids.

of the computation is distributed by calculating the bid prices locally and the group is automatically formed for a person by choosing the cameras with top bidding prices.

6.1.1. Background

An auction is the process of selling an item (goods or services) from the auctioneer to many potential buyers, i.e., bidders. Typically, in the auction, the potential buyers first offer their prices (the price offer is also called a bid) [106][107]. If the potential buyers bid for profitable trades only, we say that they are rational. Then, the auctioneer collects the bid prices information, and decides who wins the item and how much the winner has to pay. In the real world, there are many kinds of auction, which specify different bidding rules and different final payments of the winner. For example, in the first-price sealed-bid (FPSB) auction, all bidders simultaneously submit sealed bids so that no bidder knows the bid of any other participants. The bidder with the highest bid pays the submitted price. This is similar to the case in the proposed approach.

If any agent in a system cannot increase its well-being without damaging others' well-beings, we say that it is Pareto optimum [108]. The advantage of selling an item through auction method lies in the fact that in spite of asymmetric buying and selling information among bidders and auctioneer, the Pareto optimum can be achieved through auction under Revelation Principle, which rules out the possible inefficiency caused by asymmetric information [110].

6.1.2. Problem Formulation

The goal of the proposed approach is to form groups of cameras dynamically to follow multiple objects in the camera network. We want to select the cameras which have better quality of views (QOV) for an object, based on our pre-defined metrics, to form a group. This group may include the cameras which currently can "see" the object as well as those cameras which may have a high QOV by panning or tilting to somewhere else. The analogy of a real auction in economics and the grouping process in the camera network is shown in Table 6.1.

Table 6.1: The Analogy of auction in economics and camera network

Economics	Camera network
Auctioneer	Central Program
Goods	Objects
Bidder	Camera
Bid price	Camera's QOV of the object
Sale of the good	Group formation of cameras

A virtual auctioneer (a component that is not a real device like a camera, but something that is manipulated by the program) holds an auction for each of the objects in the system, i.e. objects are goods for sale. All the potentially available cameras are modeled as potential buyers for the object. There is a set of metrics according to which the cameras will evaluate their willingness to buy the good or not and if they decide to buy how much bid price they will provide. The auctioneer collects all this information and finally

makes a decision on who should sell goods, i.e., to which camera(s) to use to follow objects.

6.2. Auction Mechanism for Camera Network

6.2.1. System Assumptions

Before describing the detailed approach, we first clarify some assumptions made in our system:

1. We assume the objects to be tracked are human beings walking on a flat planar. The feet of these persons are visible so that the position of a person in one camera can be mapped to another camera with overlapping field of views (FOVs) by using homographies.

2. Homographies are calculated and the cameras' heights are known, so that we know the coordinate conversion between different camera images.

The camera's focal length is set to a fixed number such that the angle of view (the largest angle that a camera can cover without any active control) is 51.2°. Each camera has 8 overlapping pre-defined pan settings to seamlessly cover 360 degrees.

3. Also, there are three tilt settings, up 5°, down 5° (or -5°) and no tilt (0°). So, there are 24 settings for each cameras. We will call these 24 settings for Camera $C_j$as $l = \{l_j^1, l_j^2, \dots, l_j^{24}\}$ where l_j^1 is the current location of Camera C_j.

4. The cameras are rational and honest, i.e. they calculate their bid price solely based on the pre-defined metrics and they will only do the profitable trades.

5. There is no communication error.

6. There is no communication congestion.

Based on the above assumptions, we propose an auction protocol to form groups of cameras automatically and dynamically to follow the objects in the network. For the convenience of the readers, some notations that are used in the following description are summarized in Table 6.2.

Table 6.2: Symbols and notations used in Chapter 6

Symbols	Notations	Symbols	Notations
P_i	Person i	ρ	A percentage number decided by the user
C_j	Camera j	N_i	The number of cameras that in the group to follow P_i
n_c	The number of cameras that can "see" P_i	M_{ijm}	The m^{th} metric score for P_i in C_j
l	Camera setting vector	w_m	Weights for different metrics
l_j^k	The k^{th} setting of C_j	γ	Threshold for the size metric
B_{ij}	Bid price sent from camera C_j for person P_i	(x, y)	Current location of the person in the camera image
L_i	Location of P_i in the leader camera	(x_c, y_c)	Center of the camera image
$\boldsymbol{b_{ij}}$	Bid vector from C_j for P_i	α_k	Weight on k^{th} dimension in bid price function
b_{ij}^k	Intermediate bid from C_j for P_i at the setting l_j^k	λ	Elasticity of substitution between different dimensions in bid price function
B_i	$B_i = B_{i1} + B_{i2} + \cdots + B_{in_C}$		

6.2.2. Auction Protocol

The auction protocol inspired by [111] is described as follows:

1. ***Task announcement.*** A virtual agent (program running on a central server) holds an auction for each object to be tracked. An auction message is broadcast to the

whole network. The message includes information such as the location of an object and camera IDs of those cameras which are in the same group to follow it. Note that we will initialize the location of the object by a motion detection module. The camera that first "sees" the object will be initialized as the *leader camera* in the group to follow this object. The object's location is initialized as the centroid location in the leader camera's image. After that, the leader camera is decided as the one with the highest bid price and the object's centroid in this leader camera will be broadcast.

2. ***Bid price calculation***. The overall bid price B_{ij}, which is from camera C_j for person P_i, is decided by a 24-dimensional bid vector, $\boldsymbol{b_{ij}} = \{b_{ij}^1, b_{ij}^2, \dots, b_{ij}^k, \dots, b_{ij}^{24}\}, k \in [1,24]$. b_{ij}^k stands for the *intermediate bid* that the camera can get by panning or tilting to the setting l_j^k. If it cannot "see" an object at l_j^k, then b_{ij}^k is 0. Otherwise, b_{ij}^k is decided by the pre-defined metrics, such as the view, size and position of the object, which will be discussed in the next subsection. The order of elements in $\boldsymbol{b_{ij}}$ implies the willingness of the camera to follow an object or not. We prefer to use a camera without any panning or tilting, since panning and tilting make some frames blurred and it takes time to have a sharp image. If an object is moving at a high speed, when the camera can have a sharp image after panning or tilting a large degree of angle, the object may already be out of the FOV again. However, the necessity of having this vector representation instead of by considering the current location l_j^1 only lies in the fact that in some cases, all the cameras that can currently "see" the object have a back or side view of the object while if we pan or tilt some camera, which is currently unavailable for this object, it will have the object's frontal view, which can provide us

more information of interest. Or, there might be the case when a camera pans or tilts to another setting, it will gain more welfare by following another object instead of continuously following the object currently assigned to it. This vector representation helps to take into account the inclination of a camera, which, therefore, avoids the drawbacks of greedy algorithms. Finally, the overall bid price B is calculated as a function of all the intermediate bids in $\boldsymbol{b_{ij}}$, i.e. $B_{ij} = f(b_{ij}^1,\ b_{ij}^2, \dots, b_{ij}^k, \dots, b_{ij}^{24})$.

This function is designed in the next subsection.

3. ***Bid submission***. After evaluating the price for each object, all the related cameras send their bid prices for the object(s). As mentioned in the assumptions, the prices must be honest and can truly imply their willingness to follow an object.

4. ***Close of auction***. Unlike in the traditional auction, where the auctioneer will sell the good to the buyer who provides the highest bid price, the virtual auctioneer in our system choose the top N_i cameras (whose bid prices are the top N_i ones) to form a group to follow an object. All the prices are sorted from high to low and then are summed up. Let $B_i = B_{i1} + B_{i2} + \cdots + B_{inc}$. N_i is the minimum number such that

$$B_{i1} + B_{i2} + \cdots + B_{iN_i} \geq \rho\% \times B_i,$$

where ρ is a parameter decided by the user.

The whole auction process is described in Figure 6.1 as a block diagram. Note that the highest computational load, the calculation of bid prices, is distributed to each camera node and, thus, done locally.

6.2.3. Optimality Discussion

Intuitively, under the assumption that the cameras are rational and honest, all the cameras report their true evaluations of the object to be tracked to the virtual auctioneer. The virtual auctioneer can, thus, obtain the maximal benefit by "selling" the item (the object to be tracked) to those cameras that have the top N_i evaluations on the object. From the cameras' viewpoint, this transaction is optimal, since the camera which has the highest evaluation wins the right to track the object. Also, from the virtual auctioneer's standpoint, it can obtain the highest "payment" from the winner. The fact that the cameras always reveal their true evaluation of the object to be tracked validates that the Pareto optimality [108] of the camera grouping system is always achievable. Also, the optimal camera group is dynamically formed by this auction-based camera grouping process.

6.2.4. Metrics and Price Function Design

For the metrics used for evaluating the bids, we mainly consider the size of the person and the position of the person in the camera image, which are described as follows:

The size of the tracked person, measured by the ratio of the number of pixels inside the bounding box of the person to that of the size of the image. Assume that γ is the threshold for the best observation, i.e. when $r = \gamma$ this criterion reaches its peak value, where $r = \frac{\#\ of\ pixels\ inside\ the\ bounding\ box}{\#\ of\ pixels\ in\ the\ image\ plane}$.

$$M_{ij1} = \begin{cases} \frac{1}{\gamma} r, & when\ r < \gamma \\ \frac{1-r}{1-\lambda}, & when\ r \geq \gamma \end{cases} \quad (1)$$

The position of the person in the FOV of a camera. It is measured by the Euclidean distance that a person is away from the center of the image

$$M_{ij2} = \frac{\sqrt{(x-x_c)^2+(y-y_c)^2}}{\frac{1}{2}\sqrt{x_c{}^2+y_c{}^2}} \tag{2}$$

where (x, y) is the current position of the person and (x_c, y_c) is the center of the camera image plane.

Each intermediate bid b_{ij}^k is decided by the above metrics and is calculated

$$b_{ij}^k = \sum_{m=1}^{2} w_m M_{ijm} \tag{3}$$

where w_m is the weight for different metrics. The calculation of these M_{ijm} is described in the experimental part.

The final bid price B_{ij} is computed as

$$B_{ij} = \left(\alpha_1 (b_{ij}^1)^\lambda \ + \alpha_2 (b_{ij}^2)^\lambda + \cdots + \alpha_{24} (b_{ij}^{24})^\lambda\right)^{\frac{1}{\lambda}} \tag{4}$$

where $\alpha_1 + \alpha_2 + \cdots + \alpha_{24} = 1, \lambda \in (-\infty, +\infty)$.

This function is also known as the Constant Elasticity of Substitution (CES) in economics [108]. In economics, the CES function is proposed by the Stanford group around Arrow, Chenery, Minhas, and Solow in 1961 [110] as a generalization of the Cobb-Douglas function that allows for any (non-negative constant) elasticity of substitution. The CES function refers to a particular type of aggregator function which combines two or more types of consumption, or two or more types of productive inputs into an aggregate quantity. In recent years, the CES function has gained importance in macroeconomics and growth theory.

Although the bid price function B_{ij} can be picked up arbitrarily, we choose the form of the CES function, equation (4), mainly because it has simple explanations of parameters, and also it provides considerable flexibilities in parameterization. Intuitively, we can use the CES function to model the willingness of a camera to be panned or tilted. As discussed below, the parameter λ models how different a person is in one camera setting compared to another one. The parameter α models how one camera setting is preferred over another one. Overall, the bid price function B_{ij} implies the utility that Camera C_j would obtain if it is assigned to follow Person P_i.

The parameter λ in equation (4) measures the degree of easiness in substitution among different dimensions in the intermediate bid vector $\boldsymbol{b_{ij}}$, i.e., when multiple setting of a camera can cover the object to be followed, to what extent we can use one of these available settings to substitute among one another in terms of the cost and benefit the camera can get. Figure 6.2 depicts the contour curves of the bidding function given different λ. For the purpose of illustration, the dimension of the intermediate bid is reduced to two (i.e. each camera has only two settings), which reduces the bid price function to

$$B_{ij} = \left(\alpha_1 (b_{ij}^1)^{\lambda} \ + \alpha_2 (b_{ij}^2)^{\lambda}\right)^{\frac{1}{\lambda}} \tag{5}$$

As λ approaches to negative infinity we have

$$\lim_{\lambda \to -\infty} B_{ij} = \min\{\alpha_1 b_{ij}^1, \alpha_2 b_{ij}^2\}$$

which means that B_{ij} is determined by the $\alpha_k b_{ij}^k$ with the *lowest* value, and the change of other b_{ij}^k cannot change the final bid B_{ij}, i.e. the camera's bid price solely depends on

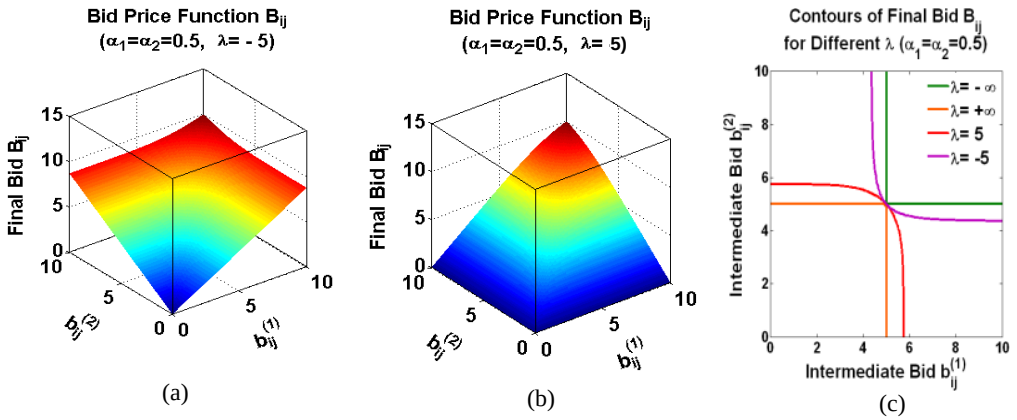
the setting that will give the worst result. Therefore, each dimension b_{ij}^k in the intermediate bid vector $\boldsymbol{b_{ij}}$ cannot be substituted by any other dimension, as shown by the green curve in Figure 6.2 (c). On the other hand, if λ equals 1, the bid price function degenerates to a simple linear function

$$B_{ij} = \alpha_1 b_{ij}^1 \ + \alpha_2 b_{ij}^2 \tag{6}$$

which means that each dimension b_{ij}^k is a perfect substitution for any other dimension in $\boldsymbol{b_{ij}}$, i.e. each setting of the camera will give exactly the same result. Finally, as λ goes to positive infinity, the bid price function converges to the *max* function

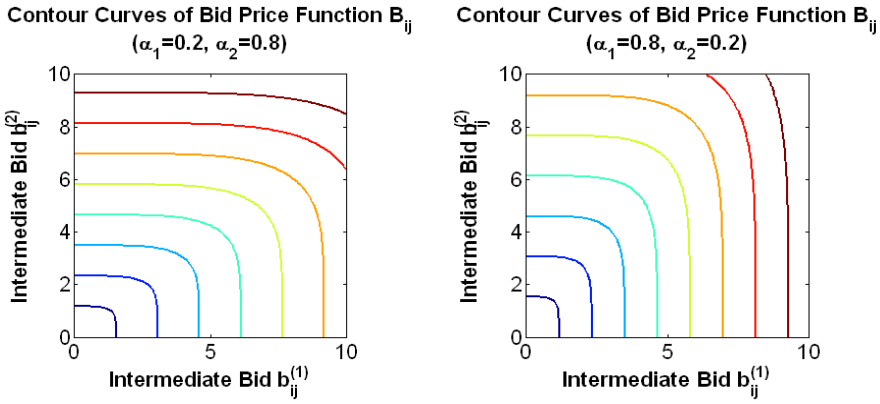
$$\lim_{\lambda \to +\infty} B_{ij} = \max\{\alpha_1 b_{ij}^1, \alpha_2 b_{ij}^2\}$$

which means that the bidder's utility level is determined by $\alpha_k b_{ij}^k$ with the highest value, and the change of other elements in b_{ij} cannot change the overall bid B_{ij}, i.e., B_{ij} solely depends on the setting that can provide the best result, as shown by the orange curve in Figure 6.2 (c). The magenta and red curves in Figure 6.2 (c) show the contours of B_{ij} for



(a) (b) (c)

Figure 6.2: Effects of different λ on the bid price B_{ij} for the case when there are only two intermediate bids. Bid Price increases as the color changes from blue to red. (a) Final bid price B_{ij} when $\lambda = -5$ (b) Final bid price B_{ij} when $\lambda = 5$. The extreme case for $\lambda = -\infty$ and $\lambda = \infty$ are shown in (c). In all these figures, α_1 and α_2 are fixed to 0.5.



(a) (b)

Figure 6.3: Contour Curves of B_{ij} (The effect of different α_k on the final bid price B_{ij} with fixed λ.) Bid Price associated with contour curve increases as the color changes from blue to red.)

two example cases that when $\lambda \in (-\infty, +\infty)$ and their actual functions are shown in Figure 6.2 (a) and Figure 6.2 (b) respectively. Intuitively, in the camera network scenario, it makes no sense to make $\lambda < 0$, since we will never use a camera with the worst result to follow an object. On the other hand, when a camera has more than one setting that can "see" an object, these settings will not be perfect substitution for one another, since panning or tilting the camera with different angles may cause different time delay and blur the image. In the parameterization of λ, we prefer the range $(1, +\infty)$, which means the camera's bid price B_{ij} depends largely on higher intermediate bids other than those lower ones.

In addition, α_k in the bid price function measures the camera's relative preference on b_{ij}^k to other b_{ij}^n $(n \neq k)$. The larger the α_k is, the larger weight is put on b_{ij}^m in the bid price function B_{ij}. One extreme case is $\alpha_k = 1$, then the bid price function degenerates to $B_{ij} = b_{ij}^k$, which means that only b_{ij}^k contributes to the utility of camera C_j in following

person P_i. Figure 6.3 describes the contour curves of B_{ij}bidding function under different parameterizations on α_i (the dimension of the intermediate bid is reduced to two for the convenience of illustration). In our experiments, we put the highest weight on α_1, which means that we prefer to use a camera to follow a person without any active control to avoid blurred images.

Note that if the dimensions of the camera setting vector l are non-overlapped with each other, then there is only one non-zero dimension in the bid vector b_{ij}. Thus, the bid vector simplifies to a scalar.

The zoom control is done when a person's frontal view is detected around the centroid of an assigned camera. We zoom in that camera (if more than one are available for the frontal view, then we zoom in the one that provides a higher bid) for 2 seconds and then zoom out to the original setting(in case that some other person will be lost when zooming in the camera).

6.3. Experiments

6.3.1. Data and Parameters

We perform the experiments in our department building, where we have 37 outdoor cameras in a network and several movable cameras to put anywhere indoor. All the cameras are commercially available Axis 215 PTZ cameras. The map of the camera network is given in Figure 6.4. We evaluate the proposed approach in several experimental cases, both indoor and outdoor, good and poor lighting conditions, different numbers of cameras and persons. All these cases are listed in Table 6.3.

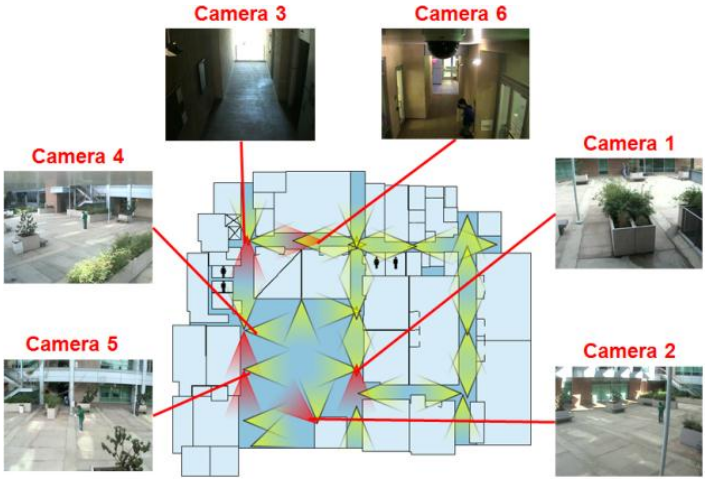


Figure 6.4: Map of the camera network.

Table 6.3: Experimental cases

Cases	No. of Cameras	No. of Persons	Lengths
Case 1	3	2	858
Case 2	6	4	1962
Case 3	6	6	2928

We calculate the homographies for different settings of cameras such that we know the correspondence between each pair of cameras for any setting. The homographies are computed based on the same ground plane off-line. When a person's location in the image of the lead camera is known, we can use the homographies to predict the person's location in all possibly available settings for all the *potentially available* cameras. Then the height of this person is estimated by using homography and the height of camera placement, which is measured beforehand. Using the camera height, we are able to estimate the person's actual height in the world coordinates from his height in pixel in the image of lead camera. Similarly, we can estimate the person's height in

pixel in all those potentially available camera settings (from those cameras that cannot "see" the person currently but it is possible to "see" this person by panning or tilting) from the previously estimated actual height in world coordinates. M_{ij1}, the size metric of the person, is estimated by making a bounding box using the same ratio of height to width of a person as it is in the lead camera and calculate the area of the bounding box. M_{ij2}, the position metric of the person, is estimated by picking the center of the top and bottom (deduced from the height) as the centroid of the person. Error occurs when the person is not fully visible. The inaccuracy is caused by the inaccuracy in the measurement in the world coordinates and the assumption for the flat ground.

We apply the online boosting tracker [57] to do tracking. This tracker uses Haar-like wavelets, orientation histograms, and local binary patterns (LBP) as the features. These features are insensitive to illumination changes, makes the tracker robust under different illumination conditions. We modify this tracker a little bit so that the size of the bounding box can change accordingly. The face detection is done by applying the face detector in OpenCV around the top half of the bounding box. We choose a particle filter tracker because it is relatively robust to occlusions. It is to be noted that the focus of this chapter is not to design a robust tracker and face detector, but lies in how to form groups of cameras dynamically and integrate camera active control into this process.

The parameters in the experiments are set empirically. The threshold for the size of the person is $\gamma = \frac{1}{15}$. The weights for different metrics are selected as $w_1 = 0.6$ and $w_2 = 0.4$. The weights in the bid price function B_{ij} are given in Table 6.4.

From values of α_k, we can note that using the camera with active control as little as possible is preferred, since it may cause blurring of images and the time delay may cause missing more objects. The elasticity of substitution parameter in the bid price function B_{ij}, λ =8. The percentage based on which we decide the number of cameras to form a group, ρ =50.

Table 6.4: Value of α_k

α_k	Pan	Tilt	Value	α_k	Pan	Tilt	Value
α_1	0°	0°	0.150	α_{13}	-90°	5°	0.020
α_2	45°	0°	0.080	α_{14}	135°	5°	0.010
α_3	-45°	0°	0.080	α_{15}	-135°	5°	0.010
α_4	90°	0°	0.030	α_{16}	180°	5°	0.006
α_5	-90°	0°	0.030	α_{17}	0°	-5°	0.100
α_6	135°	0°	0.015	α_{18}	45°	-5°	0.060
α_7	-135°	0°	0.015	α_{19}	-45°	-5°	0.060
α_8	180°	0°	0.010	α_{20}	90°	-5°	0.020
α_9	0°	5°	0.100	α_{21}	-90°	-5°	0.020
α_{10}	45°	5°	0.060	α_{22}	135°	-5°	0.010
α_{11}	-45°	5°	0.060	α_{23}	-135°	-5°	0.010
α_{12}	90°	5°	0.020	α_{24}	180°	-5°	0.006

6.3.2. Error Metrics

The proposed approach aims to solve the camera active control problem. So all the experiments have to be done in real-time and the decisions for panning or tilting a camera have to be made on-line. This makes it hard to compare the results with a pre-calculated ground-truth data, or with others' results. Since we use natural human beings to perform the experiments, there is no way to repeat the experiments exactly the same in several trials. What we do is to pre-define the paths and repeat the experiments for 10 trials and get the average result. If the overlapping between the bounding box returned by the

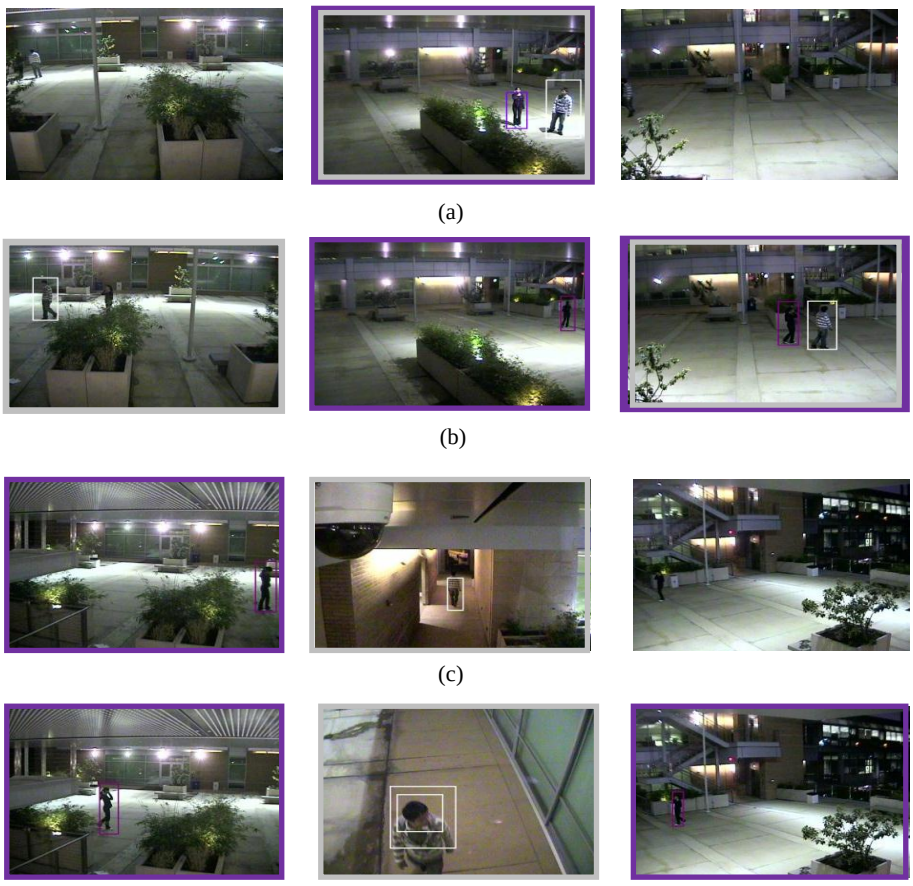


Figure 6.5: Some typical frames for the 3 cameras 2 persons case. The camera images from the cameras in the same group for a person are boxed in the same color as the person they are assigned to follow.

tracker and the object is more than 70% and less than 150%, plus there is no other camera with a better QOV, then it is correct. Otherwise, we say this is an error frame.

6.3.3. Experimental Results

In Figure 6.5, we show some typical frames in a simple case where we deploy 3 cameras (Camera 1, Camera 4 and Camera 5) and let 2 persons walk in the camera network. In frame (a), although all the three cameras that can "see" the person in grey, camera 2's bid price takes up to 67% of the summation of all the bids. Therefore, there is only Camera 2

in the group that is assigned to follow the person in grey. In frame (b), although Camera 2 is potentially available for both of the two persons, it can "see" the person in purple pretty well without panning or tilting, thus, its bid for this person is higher and it forms a group with Camera 3 together for the person in purple while the person in grey is monitored by the group formed by Camera 1 and Camera 3. In frame (c), the frontal view of the person in red is detected in camera 3. So, we zoom (1.5 times) in camera 3 to have a close-up view. In frame (d), the person in grey can only be covered by camera 2. The process of bidding for the person in grey (Person 1) is shown in Figure 6.6.

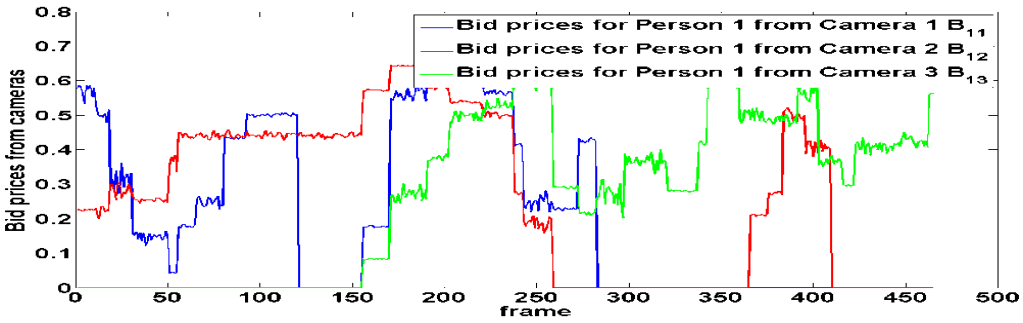


Figure 6.6: Bid prices for the person in grey (Person 1). See Figure 5 for the grouping results for the person.

In Figure 6.7, we show some typical frames in a more complicated case with 6 cameras and 4 persons. As stated previously, QOVs in the cameras influence the proposed camera grouping results. We can observe that, when there are more than one camera available for a person and none of them can dominate any other in terms of the tracking quality, then these cameras will form a group to follow this person, e.g. in frame a for the person in red. Otherwise, when there is a camera that has a much higher score for a person, then it's bid price will be greater than 50% of the overall summation of all

Figure 6.7: Experimental results in the 6 cameras 4 persons case. The camera images from the cameras in the same group for a person are boxed in the same color as the person they are assigned to follow.

Figure 6.8: Experimental results in the 6 cameras 6 persons case. The camera images from the cameras in the same group for a person are boxed in the same color as the person they are assigned to follow.

Camera 1	**Camera 2**	**Camera 3**
Camera 4	**Camera 5**	**Camera 6**

the bids. In this case, e.g., in frame (c) for the person in green, the camera can provide us with much information so that we just use it to follow the person. This includes the case when there is only one camera that is available for a person, e.g., in frame (c).

We show some example frames for the 6 cameras 6 persons case in Figure 6.8. In frame (a), Camera 1 and Camera 2 forms one group to follow the person in blue bounding box, while the other cameras are used to follow other persons, with a single camera in a group. Since all the persons currently appearing in the network are covered, Camera 2 is panned to get a better view for that person in blue bounding box in frame (b). In Frame (c), 4 persons (the persons in red, yellow, blue and black bounding boxes) are currently visible in Camera 5. Camera 5 is in the group for following the persons in black and blue bounding boxes. The person in black bounding box is about to leave the FOV of Camera 5, and since Camera 2 cannot get a better view for this person (it is far from the person), Camera 5 is panned in frame (d) for that person. This make Camera 5 lose the view for the person in red, but it is okay since Camera 1 is covering that person in frame (d).Similarly in frame (e) - frame (h), the cameras form into different groups for different persons, as shown in the figure by different colors. When a face is detected in frame (e) and frame (g) in Camera 2, it is zoomed in to acquire a higher resolution view of that person, as shown in frame (f) and frame (h).

The overall performance of the proposed approach is shown in Table 6.5. By using the error metrics defined in the Section 6.3.2, we calculate the correction rates in very experimental trial and compute the average results based on these results. Case 1 has

Table 6.5: Correction rates in different experimental cases

Cases	Trial 1 (%)	Trial 2 (%)	Trial 3 (%)	Trial 4 (%)	Trial 5 (%)	Trial 6 (%)	Trail 7 (%)	Trial 8 (%)	Trial 9 (%)	Trail 10(%)	Avrg. (std)
Case 1	89.79	87.88	92.96	77.77	89.98	90.11	92.33	89.62	92.34	88.94	89.17 (4.32)
Case 2	94.32	93.25	91.78	89.66	95.67	96.23	88.45	90.37	91.28	93.84	92.49 (2.60)
Case 3	90.45	89.92	88.33	93.33	91.31	87.96	92.56	91.45	93.79	92.88	91.20 (2.03)

Table 6.6: Correction rates in case 3 by using different approaches

Appr.	Trial 1 (%)	Trial 2 (%)	Trial 3 (%)	Trial 4 (%)	Trial 5 (%)	Trial 6 (%)	Trail 7 (%)	Trial 8 (%)	Trial 9 (%)	Trail 10(%)	Avrg. (std)
Qureshi et al. 2007	86.97	90.23	84.45	87.77	89.96	88.63	90.76	87.45	90.12	84.11	88.05 (2.37)
The proposed approach	90.45	89.92	88.33	93.33	91.31	87.96	92.56	91.45	93.79	92.88	91.20 (2.03)

a lower correction rate because in that case, the tracker gets lost easily because of the noisy image obtained during the evening time. We show a comparison of our proposed approach with the approach in [98] in Table 6.6. It can be observed that in most trials the proposed approach get a better result. In the two trials (Trial 2 and Trial 6), where the proposed approach is worse, the differences between the results are small, less than 1%.

6.4. Summary

We proposed a novel auction-based mechanism to form groups of cameras to follow objects in a camera network. This chapter introduced the auction concept into the camera network area and achieved promising results. A virtual auctioneer holds an auction for each object to be followed in the network. Bid prices are calculated locally so that the computation is distributed a lot. At the meantime, the final decision is made by the central virtual auctioneer, and thus can make sure to get the global Pareto optimum. In

the auction protocol design, we made the bid price as a vector representation, to take into account the cameras' willingness to follow an object or not. By doing so, plus the help of the pre-calculated homographies, we can also consider to pan or tilt some cameras to get a better view of the object even if the object is currently invisible in these cameras. We provide some intuitive design for the bid price computation as well, which are easy to observe and evaluate. However, it is to be noted that these criteria are subject to the user. The user can provide different kinds of criteria to meet different requirements under different surveillance scenarios.

We show results for following various number of persons, active control of cameras and dynamic group formation in several experimental cases. These experiments are performed in real-time, under different environmental conditions. By deploying multithreading techniques, the data can be processed at a frame rate of 15-20 fps. We show the effectiveness of the proposed approach and also compare the proposed approach with other state-of-the-art work, which also validates the proposed approach.

Chapter 7

Conclusions and Future Work

In this study, we propose to introduce a series of economic models into the camera selection, handoff and control problem. These include the potential game approach which we describe in Chapter 3, the weakly acyclic game approach in Chapter 4, and the auction-based camera active control in Chapter 6. Because all these approaches rely on a robust tracker, in Chapter 5, we propose to combine the camera selection process with the fusion of multiple trackers.

In Chapter 3, we proposed a new principled approach based on game theory for the camera selection and handoff problem. We developed a set of intuitive criteria in this chapter and compared them with each other as well as the combination of them. Our experiments showed that the combined criterion is the best based on the error definition provided in Chapter 3. Since the utilities, input of the bargaining process, largely depend on the user-supplied criteria, our proposed approach can be task-oriented. Unlike the conventional approaches which perform camera handoffs only when an object is leaving or entering the FOV, we can select the "best" camera based on the pre-defined criteria.

The key merit of the proposed approach is that we use a theoretically sound game theory framework with bargaining mechanism for camera selection in a video network so that we can obtain a stable solution with a reasonably small number of iterations. The approach is independent of (a) the spatial and geometrical relationships among the cameras, and (b) the trajectories of the objects in the system. It is robust with respect to

multiple user-supplied criteria. The approach is flexible since there is no requirement for a specific criterion that a user is obligated to use. A wide variety of experiments show that our approach is computationally more efficient and robust with respect to other existing approaches.

We analyzed the influence of a tracker on the proposed approach and compared our work with two other recent approaches both qualitatively and quantitatively. All the experiments used a physical camera network with real data in real time. This included both indoor and outdoor environments with different numbers of cameras and persons. As compared to the other approaches, it is shown that the proposed approach has smaller error rates in all the experiments. The computational efficiency of the proposed approach is also verified quantitatively. This comparison shows that (a) COR approach cannot do any criterion-dependent camera selection. (b) As the number of cameras and persons in the system increases, the selection ambiguity and failure also increase in the COR approach. (c) The CSP approach is task-dependent and can select the "best" camera based on whatever criterion is provided by the user. (d) The CSP approach is computationally much more expensive than our approach.

In Chapter 4, we model the camera selection and handoff problem as a weakly acyclic game and use the payoff based learning algorithm to get the stable result with guaranteed convergence. We develop more criteria so that the handoffs occur in a smooth manner and take time delay for awakening a camera into consideration. We compare the proposed approach with the potential game approach, both theoretically and experimentally. This comparison shows that the weakly acyclic game approach is much

more efficient than the potential game approach. Further, the weakly acyclic game approach removes the requirement of alignment of local and global utilities needed in the potential game approach. So, in the weakly acyclic game approach, the design of criteria and the payoff function for cameras are both more flexible and easier. Since no global information is needed to carry out the camera selection and hand-off in the weakly acyclic game approach, the system is realized in a distributed manner. We show results with real data in 6 different cases, both indoors and outdoors, with different numbers of cameras and persons. We also compared related non-game theoretic approaches [25]. All the results show the efficacy of the proposed approach.

In Chapter 5, we proposed an approach to do camera selection using a score-level fusion of multiple state-of-the-art trackers with a novel tracker re-initialization scheme. The fusion process does not only take into account a tracker's accuracy, but also the tracked object's attributes according to camera selection criteria. A tracker with a higher tracking performance but worse camera selection attributes may be avoided in the final selection. In this way, the fusion of multiple trackers and the camera selection are done at the same time. To benefit from the combination of fusion of multiple trackers and camera selection, the selected tracker's information is used to re-initialize the locations of other trackers. This is achieved by making use of the pre-calculated homography and epipolar geometry together. A feet test is performed before this to make the re-initialization more reliable.

The proposed approach is evaluated using several widely used public datasets. The effectiveness of the proposed approach is shown by providing many comparisons.

We show the necessity of applying multiple trackers than using a single tracker only and the necessity of doing scene analysis with feet existence test and the epipolar geometry. The proposed joint optimization approach provides robust tracking and camera selection: 1) The closed-loop framework with feedback from the camera selection results to re-initialize the trackers improves the overall performance by 9.88% on the average. 2) The feet test before camera selection feedback improves the overall performance by 3.91% on the average; 3) The usage of epipolar geometry improves the overall performance by 3.02% on the average; 4) Fusion of multiple trackers, comparing with individual trackers, improves the overall results by 10.1% on the average.

In the future, this work will be integrated with automatic human detection algorithms to do automatic initialization. We will also include more complex criteria to make the system more flexible.

In Chapter 6, we proposed a novel auction-based mechanism to form groups of cameras to follow objects in a camera network. This chapter introduced the auction concept into the camera network area and achieved promising results. A virtual auctioneer holds an auction for each object to be followed in the network. Bid prices are calculated locally so that the computation is distributed a lot. At the meantime, the final decision is made by the central virtual auctioneer, and thus can make sure to get the global Pareto optimum. In the auction protocol design, we made the bid price as a vector representation, to take into account the cameras' willingness to follow an object or not. By doing so, plus the help of the pre-calculated homographies, we can also consider to pan or tilt some cameras to get a better view of the object even if the object is currently

invisible in these cameras. We provide some intuitive design for the bid price computation as well, which are easy to observe and evaluate. However, it is to be noted that these criteria are subject to the user. The user can provide different kinds of criteria to meet different requirements under different surveillance scenarios.

We show results for following various number of persons, active control of cameras and dynamic group formation in several experimental cases. These experiments are performed in real-time, under different environmental conditions. By deploying multithreading techniques, the data can be processed at a frame rate of 15-20 fps. We show the effectiveness of the proposed approach and also compare the proposed approach with other state-of-the-art work, which also validates the proposed approach.

Acknowledgement

This work was partially supported by NSF grants 0551741, 0622176, 0905671 and ONR grants (DoD Instrumentation and the Aware Building).

Bibliography

[1] O. Javed, S. Khan, Z. Rasheed and M. Shah. Camera Hand-off: Tracking in Multiple Uncalibrated Stationary Cameras. *IEEE Workshop on Human Motion*, 2000, pp. 113-118.

[2] J. Park, P. C. Bhat, and A. C. Kak. A Look-up Table Based Approach for Solving the Camera Selection Problem in Large Camera Networks. *Int'l Workshop on Distributed Smart Cameras*, 2006.

[3] Y. Jo and J. Han. A New Approach to Camera Hand-off without Camera Calibration for the General Scene with Non-planar ground. *ACM Int'l Workshop on Video Surveillance and Sensor Networks*, 2006, pp. 195-202.

[4] F. Z. Qureshi and D. Terzopoulos. Multi-Camera Control through Constraint Satisfaction for Pedestrian Surveillance. *AVSS 2008*, pp. 211-218.

[5] X. Zou and B. Bhanu. Anomalous Activity Classification in the Distributed Camera Network. *ICIP*, 2008, pp. 781-784.

[6] V. Kettnaker and R. Zabih. Bayesian Multi-camera Surveillance. *CVPR*, 1999, vol. 2, pp. 253-259.

[7] T. Chang and S. Gong. Bayesian Modality Fusion for Tracking Multiple People with a Multi-Camera System. *European Workshop on Advanced Video-based Surveillance Systems*, 2001.

[8] J. Kim amd D. Kim. Probabilistic Camera Hand-off for Visual Surveillance. *ICDSC*, 2008.

[9] J. Kang, I. Cohen, and G. Medioni. Continuous Tracking within and across Camera Streams. *CVPR*, 2003, pp. 267-272.

[10] O. Javed, S. Khan, and M. Shah. Appearance Modeling for Tracking in Multiple Non-overlapping Cameras. *CVPR*, 2005, pp. 26-33.

[11] B. Song, A. Roy-Chowdhury. Stochastic Adaptive Tracking In A Camera Network. *ICCV*, 2007.

[12] Q. Cai and J. K. Aggarwal. Tracking Human Motion in Structured Environments Using a Distributed Camera System. *PAMI*, 1999, pp. 1241-1247.

[13] G. Arslan, J. R. Marden and J. S. Shamma. Autonomous Vehicle-Target Assignment: A Game-Theoretical Formulation. *ASME Transactions on Dynamic Systems, Measurement, and Control*, special issue on *Analysis and Control of Multi-Agent Dynamic Systems*, 2007, vol. 129, no. 5, pp. 584-596.

[14] L. Tessens, M. Morbee, H. Lee, W. Philips, and H. Aghajan, "Principal view determination for camera selection in distributed smart camera networks," in *2nd ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* pp. 1 –10, Sept. 2008.

[15] R. B. Myerson. Game Theory-Analysis of Conflict. Harvard University Press, Cambridge, MA, 1991.

[16] M. J. Osborne. An Introduction to Game Theory. Oxford University Press, 2003.

[17] http://plato.stanford.edu/entries/game-theory/#Uti

[18] G. R. Bradski. Computer Vision Face Tracking for Use in a Perceptual User Interface. Intel Technology Journal Q2, 1998.

[19] http://opencv.willowgarage.com/wiki/CvReference

[20] M.J. Osborne and A. Rubinstein. A Course in Game Theory. The MIT Press, Cambridge, MA, 1994.

[21] B. Bhanu, C.V. Ravishankar, A.K. Roy-Chowdhury, D. Terzopoulos and H. Aghajan, (Eds.), "Distributed Video Sensor Networks," *Springer* 2011.

[22] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, Dec. 2006.

[23] W. Brendel, M. Amer and S. Todorovic, "Multiobject tracking as maximum independent set," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp.1273-1280, Jun. 2011.

[24] A. Andriyenko, K., Schindler and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Jun. 2012.

[25] K. Morioka, S. Kovacs, J.-H. Lee, P. Korondi. A cooperative object tracking system with fuzzy-based adaptive camera selection" *International Journal on Smart Sensing and Intelligent Systems*, Vol. 3 (3), Sept. 2010.

[26] Y. Li and B. Bhanu, "Utility-based camera assignment in a video network: A game theoretic framework," in *IEEE Sensors Journal,* Issue 3, pp. 676-687, Mar. 2011.

[27] Y. Li and B. Bhanu, "Task-oriented Camera Assignment Approaches", *IEEE International Conference on Image Processing*, Cairo, Egypt, Jul. 2009.

[28] Y. Li, B. Bhanu, and W. Lin, "Auction protocol for camera active control", *IEEE International Conference on Image Processing*, Hong Kong, China, Sept. 2010.

[29] Y. Li and B. Bhanu, "Fusion of multiple trackers in video networks", *IEEE/ACM International Conference on Distributed Smart Cameras*, Gent, Belgium, Aug. 2011.

[30] Y. Li, B. Bhanu, and V. Nguyen, "On the performance of handoff and tracking in a camera network", *IEEE International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 2010.

[31] Y. Li and B. Bhanu, "A comparison for techniques for camera selection and hand-off in a video network", *IEEE/ACM International Conference on Distributed Smart Cameras*, Como, Itly, Aug. 2009.

[32] M. Quaritsch, M. Kreuzthaler, B. Rinner, H. Bischof, and B. Strobl, "Autonomous multicamera tracking on embedded smart cameras," *EURASIP Journal on Embedded Systems*, pp. 1–11, Feb. 2007.

[33] S. Fleck and W. Strasser, "Adaptive probabilistic tracking embedded in a smart camera," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshop,* p. 134, Jun. 2005.

[34] C. Micheloni, G. Foresti, and L. Snidaro, "A network of co-operative cameras for visual surveillance," in *IEE Proceedings on Vision, Image and Signal Processing,* vol. 152, pp. 205 – 212, Apr. 2005.

[35] 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 *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshop,* p. 118, Jun. 2006.

[36] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi, "Camera handoff with adaptive resource management for multicamera multi-target surveillance," in *IEEE Fifth Intl. Conf. on Advanced Video and Signal Based Surveillance,* pp. 79 –86, Sept. 2008.

[37] J. Nayak, L. Gonzalez-Argueta, B. Song, A. Roy-Chowdhury, and E. Tuncel, "Multi-target tracking through opportunistic camera control in a resource constrained multimodal sensor network," in *2nd ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* pp. 1 –10, Sept. 2008.

[38] B. Horling, R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins, and V. Lesser, "Distributed sensor network for real time tracking," in *5th Intl. Conf. on Autonomous Agents*, pp. 417–424, Jun. 2001.

[39] F. Qureshi and D. Terzopoulos, "Smart camera networks in virtual reality," *Proceedings of the IEEE*, vol. 96, pp. 1640 –1656, Oct. 2008.

[40] B. Song, C. Soto, A. Roy-Chowdhury, and J. Farrell, "Decentralized camera network control using game theory," in *2nd ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* pp. 1 –8, Sept. 2008.

[41] S.-N. Lim, A. Elgammal, and L. Davis, "Image-based pan-tilt camera control in a multi-camera surveillance environment," in *Multimedia and Expo, Intl. Conf. on*, vol. 1, pp. I – 645–8, Jul. 2003.

[42] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff based dynamics for multi-player weakly acyclic games," in *SIAM Journal on Control and Optimization, Special Issue on Control and Optimization in Cooperative Networks*, 2007.

[43] N. Ukita, "Real-time cooperative multi-target tracking by dense communication among active vision agents.," *Web Intelligence and Agent Systems*, pp. 15–29, 2007.

[44] M. Trivedi, H. Kohsia, and I. Mikic, "Intelligent environments and active camera networks," in *Systems, Man, and Cybernetics, IEEE Intl. Conf. on*, vol. 2, pp. 804 – 809, 2000.

[45] C-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan and M. Abidi, "Camera handoff and placement for automated tracking system with multiple omnidirectional cameras," Computer Vision and Image Understanding, Vol. 114, Issue 2, Feb. 2010.

[46] K. Nummiaro, E. Koller-meier, T. Svoboda, D. Roth, and L. V. Gool, "Color-based object tracking in multi-camera environments," in *DAGM 25th Pattern Recognition Symposium*, pp. 591–599, 2003.

[47] P. C. M. Huang and R. Malhame, "Distributed multi-agent decision making with partial observations: Asymptotic Nash equilibrium," in *17th International Symposium on Math. Theory on Networks and Systems*, pp. 2725–2730, Jul. 2006.

[48] K. Ghoul and A. Khababa, "Multi agent system and Preto optimal auctions," *International Journal of Computer Applications,* Article 2, Nov. 2011.

[49] J. Josephson, "Stochastic better-reply dynamics in finite games," *Economic Theory*, vol. 35, pp. 381–389, 2008.

[50] H. Nguyen, B. Bhanu, A. Patel, and R. Diaz, "Videoweb: Design of a wireless camera network for real-time monitoring of activities," in *3rd ACM/IEEE Intl. Conf. on Distributed Smart Cameras,* pp. 1 –8, Sept. 2009.

[51] Y. Li, B. Bhanu, and V. Nguyen, "On the performance of handoff and tracking in a camera network," in *20th Intl. Conf. on Pattern Recognition,* pp. 3645 –3648, Aug. 2010.

[52] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Intl. Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[53] http://viper-toolkit.sourceforge.net/docs/gt/.

[54] "Cisco IP Video Surveillance Design Guide", http://www.cisco.com/en/US/docs/solutions/Enterprise/Video/IPVS/IPVS_DG/IPVS-DesignGuide.html

[55] R. Y. Tsai ,"A Versatile Camera Calibration Technique for 3D Machine Vision", *IEEE J. Robotics & Automation*, RA-3, No. 4, pp. 323-344, Aug. 1987,

[56] X. Wang, T.X. Han and S. Yan, "An HOG-LBP human dtector with partial occlusion handling," *IEEE Intl. Conf. on Computer Vision*, 2009.

[57] H. Grabner, M. Grabner and H. Bishof. Online Boosting and Vision. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'06), Vol. 1. IEEE Computer Society, Washington, DC, USA, 260-267.

[58] H. Grabner, C. Leistner and H. Bishof. Semi-supervised online boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part*

I (ECCV'08), David Forsyth, Philip Torr, and Andrew Zisserman (Eds.). Springer-Verlag, Berlin, Heidelberg, 234-247.

[59] B. Babenko, M. Yang and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Analysis And Machine Intelligence* (PAMI'11)33, 8 (August 2011), 1619-1632.

[60] K. Bernardin and R. Stiefelhagen. Audio-visual multi-person tracking and identification for smart environments. In *Proceedings of the 15th International Conference on Multimedia* (MULTIMEDIA '07). ACM, New York, NY, USA, 661-670.

[61] C. Conaire, N. O'Connor and A. Smeaton. Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. *Mach. Vision Appl.* 19, 5-6 (September 2008), 483-494.

[62] T. Cham and J. Rehg. Multi-modal tracking using texture changes. *Image Vision Computing*, 26, 3 (March 2008), 442-450.

[63] S.M. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of the 9th European Conference on Computer Vision* (ECCV '06). Graz, Austria, 133-146.

[64] R. Eshel and Y. Moses. Tracking in a Dense Crowd Using Multiple Cameras. *International Journal on Computer Vision* (IJCV'10) 88, 1 (May 2010), 129-143.

[65] M. Taj and A. Cavallar. Multi-camera track-before-detect. In *Proceedings of the 3rd International Conference on Distributed Smart Cameras* (ICDSC'09). ACM/IEEE, Como, Italy, 1-6.

[66] B. Kwolek. Multi Camera-Based Person Tracking Using Region Covariance and Homography Constraint. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS '10). IEEE Computer Society, Washington, DC, USA, 294-299.

[67] B. Mccane, B. Galvin and K. Novins. Algorithmic fusion for more robust feature tracking. *International Journal of Computer Vision* (IJCV'02). 49, 1, 79-89.

[68] B. Zhong, H. Yao, S. Chen, R. Ji, X. Yuan, S. Liu and W. Gao. Visual tracking via weakly supervised learning from multiple imperfect oracles. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'10). IEEE Computer Society, San Francisco, CA, USA, 1323-1330.

[69] I. Leichter, M. Lindenbaum and E. Rivlin. A General Framework for Combining Visual Trackers --- The "Black Boxes" Approach. *International Journal on Computer Vision* (IJCV'06) 67, 3 (May 2006), 343-363.

[70] N.T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *Proceedings of the 7th European Conference on Computer Vision-*

Part IV (ECCV '02), Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (Eds.). Springer-Verlag, London, UK, UK, 373-387.

[71] C. Leistner, P. Rpth, H. Grabner, H. Bischof, A. Starzacher and B. Rinner. Visual online learning in distributed camera networks. In *Proceedings of the 2nd International Conference on Distributed Smart Cameras* (ICDSC'08). ACM/IEEE, Palo Alto, California, 1-10.

[72] A. Tyagi and J.W. Davis. A context-based tracker switching framework. IEEE Workshop on Motion and Video Computing (WMVC'08). IEEE, Copper Mountain, CO, USA, 1-8.

[73] A. Mittal and L. Davis. Unified multi-camera detection and tracking using region-matching. In *Proceedings of the IEEE Workshop on Multi-Object Tracking (WOMOT'01)*. IEEE Computer Society, Washington, DC, USA, 3-10.

[74] E. Monary and K. Kroschel. Dynamic Sensor Selection for Single Target Tracking in Large Video Surveillance Networks. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS '10). IEEE Computer Society, Washington, DC, USA, 539-546.

[75] P. Kelly, C. Conaire, C. Kim and N. O'Connor. Automatic camera selection for activity monitoring in a multi-camera system for tennis. In *Proceedings of the 3rd International Conference on Distributed Smart Cameras* (ICDSC'09). ACM/IEEE, Como, Italy, 1-8.

[76] L. Snidaro, I. Visentini and G. Foresti. Multi-sensor Multi-cue Fusion for Object Detection in Video Surveillance. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance* (AVSS '09). IEEE Computer Society, Washington, DC, USA, 364-369.

[77] A. Gupta, A. Mittal and L. Davis. COST: An approach for camera selection and multi-object inference ordering in dynamic scenes. In *Proceedings of the 11th International Conference on Computer Vision* (ICCV'07). IEEE, Rio de Janeiro, Brazil, 1-8.

[78] P. Pahalawatta, T. Pappas and A. Katsaggelos. Optimal sensor selection for video-based target tracking in a wireless sensor network. In *Proceedings of the 11th International Conference on Image Processing* (ICIP'04). IEEE, Vol. 5, Singapore, 3073 - 3076.

[79] S. Calderara, A. Prati and R. Cucchiara. HECO: Homography and epipolar-based consistent labeling for outdoor park surveillance. *Computer Vision And Image Understanding.* 111, 1 (July 2008), 21-42.

[80] Z. Kalal, J. Matas and K. Mikolajczyk. P-N Learning: Bootstrapping binary classifiers by structural constraints. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'10), Vol. 1. IEEE Computer Society, San Francisco, CA, USA, 49-56.

[81] M. Ozuysal M. Calonder, V. Lepetit and P. Fua. Fast keypoint recognition using random ferns. *IEEE Trans. Pattern Analysis And Machine Intelligence* (PAMI'10) 32, 3 (March 2010), 448-461.

[82] B. Song, T. Jeng, E. Staudt and A.K. Roy-Chowdhury. A stochastic graph evolution framework for robust multi-target tracking. In *Proceedings of the 11th European conference on Computer vision: Part I* (ECCV'10), Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer-Verlag, Berlin, Heidelberg, 605-619.

[83] L. Snidaro, R. Niu, G. Foresti and P. Varshney. Quality-based fusion of multiple video sensors for video surveillance. In *IEEE Transaction on System, Man and Cybernetics*. IEEE, Issue 4, Vol. 37, 1044-1051.

[84] L.G. Shapiro and G.C. Stockman. Computer Vision. Prentice Hall. River, New Jersey.

[85] http://viper-toolkit.sourceforge.net/products/gt/

[86] http://www.cyberbee.com/whodunnit/foot.html

[87] http://cvlab/eplf.ch/data/pom

[88] A. Adya, P. Bahl, J. Padhye, A. Wolman and L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. In *Proceedings of the IEEE 1st International Conference on Broadnets Networks (BroadNets'04)*. IEEE, Los Alamitos, CA, 210–217.

[89] I. F. Akyildiz, T. Melodia and K.R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Netw*. 51, 4, 921–960.

[90] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. Wireless sensor networks: *A survey. Comm*. ACM 38, 4, 393–422.

[91] P. Bahl, R. Chancre and J. Dungeon. SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceeding of the 10th International Conference on Mobile Computing and Networking (MobiCom'04)*. ACM, New York, NY, 112–117.

[92] Crossbow 2008. XBOW sensor motes specifications. http://www.xbow.com.

[93] D. Culler, D. Estrin and M. Srivastava. Overview of sensor networks. *IEEE Comput*. 37, 8 (Special Issue on Sensor Networks), 41–49.

[94] Harvard CodeBlue 2008. CodeBlue: Sensor networks for medical care. http://www.eecs.harvard.edu/mdw/proj/codeblue/.

[95] A. Natarajan, M. Motani, B. De Silva, K. Yap and K.C. Chua. Investigating network architectures for body sensor networks. In *Network Architectures*, G. Whitcomb and P. Neece, Eds. Keleuven Press, Dayton, OH, 322–328.

[96] A. Tzamaloukas and J. Garcialinaaceves. Channel-hopping multiple access. Tech. Rep. I-CA2301, Department of Computer Science, University of California, Berkeley, CA.

[97] G. Zhou, J. Li, C-Y. Wan, M.D. Yarvis and J.A. Stankovic. *Body Sensor Networks*. MIT Press, Cambridge, MA.

[98] F. Qureshi and D. Terzopoulos. Distributed coalition formation in visual sensor networks: a virtual vision approach. In *Proceedings of the 3rd IEEE international conference on Distributed computing in sensor systems (DCOSS'07)*, James Aspnes, Christian Scheideler, Anish Arora, and Samuel Madden (Eds.). Springer-Verlag, Berlin, Heidelberg, 1-20.

[99] K-W. Chen, C-C. Lai, Y-P. Hung and C-S Chen. An adaptive learning method for target tracking across multiple cameras. In *Proceeding of the 21st International Conference on Computer Vision and Pattern Recognition (CVPR '08)*. IEEE, Anchorage, Alaska, 1–8.

[100] E. Monari and K. Kroschel. A knowledge-based camera selection approach for object tracking in large sensor networks. In *Proceedings of the IEEE 3rd International Conference on Distributed Smart Cameras (ICDSC'09)*. Como, Italy. 1-8.

[101] C. Shen, C. Zhang and S. Fels. A multi-camera surveillance system that estimates quality-of-view measurement. In *Proceeding of the 14th International Conference on Image Processing (ICIP '07)*. IEEE, San Antonio, Texas, 193–196.

[102] L. He and T.R. Ioerger. Task-oriented computational economic-based distributed resource allocation mechanisms for computational grids. In *Proceedings of IC-AI*. 2004, 462-468.

[103] B.P. Gerkey and M.J. Mataric. Sold!: Auction methods for multirobot coordination. In IEEE *Transactions on Robotics and Automation*. IEEE. Issue 5, 1042-296x.

[104] M.B. Dias and A. Stentz. Opportunistic optimization for market-based multirobot control. In IEEE/RSJ *International Conference on Intelligent Robots and Systems*, IEEE, 2002. 2714-2720.

[105] J. Chen, C. Zang, W. Liang and H. Yu. Auction-based dynamic coalition for single target tracking in wireless sensor networks. *In Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA'06)*, Dalian, China. 94 - 98.

[106] E. Wolfstetter. Auctions: an introduction. *Journal of economic surveys*, vol. 10(4), pages 367-420.

[107] R.P. Mcafee and J. Mcmillan. Auctions and bidding 1987. *Journal of economic literature*, Vol. 25, No. 2 pp. 699-738, 1987.

[108] H.R. Varian. Microeconomic analysis (3rd edition). *W.W.Norton & Company*, New York,1992.

[109] K.J. Arrow, B.H. Chenery, B.S. Minhas, and R.M. Solow. Capital-labor substitution and economics efficiency. The Review of Economics and Statistics, 43(3), pp. 225-250. 1961.

[110] J.G. Riley and W.F. Samuelson. 1981. Microeconomic analysis (3rd edition). *W.W.Norton & Company*, New York,1992.

[111] R.G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE TransactioSns on Computers* C-29(12), IEEE, 1104–1113.