Dynamic Scenes and Camera Networks

Marci Lenore Meingast

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-33
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-33.html

February 22, 2009

Dynamic Scenes and Camera Networks

by

Marci Lenore Meingast

B.S. (University of Illinois, Urbana-Champaign) 2001
M.S. (University of California, Berkeley) 2005

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering-Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Shankar Sastry, Chair
Professor Ruzena Bajcsy
Professor Pamela Samuelson

Fall 2008

The dissertation of Marci Lenore Meingast is approved:

Chair Date

Date

Date

University of California, Berkeley

Fall 2008

Dynamic Scenes and Camera Networks

by

Marci Lenore Meingast

Abstract

Dynamic Scenes and Camera Networks

by

Marci Lenore Meingast

Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Shankar Sastry, Chair

In recent years, camera networks have changed dramatically. These networks now incorporate more cameras than ever before and have the capability to capture and record dynamic scene. Additionally, these networks are being used in more public and uncontrolled environments. The images from the cameras provide a rich source of information about the environment and have led to an increase in camera network applications. However, there still exist a number of implementational and social issues regarding camera networks. In this dissertation, we address aspects of the issues of data correlation, data integrity, and data privacy in these networks.

In looking at data correlation, we focus on the localization of the cameras in the network. We present a method for doing automatic localization by using the dynamic scene information these networks are now able to capture and store. Since

the cameras in the network may be wide-baseline and not see similar static features, we use the dynamic scene data and detect moving objects in the scene. In an intra-camera process, we correlate the moving objects and build their trajectories within each image plane. These trajectories become the spatio-temporal features we then use in an inter-camera step by correlating them between the cameras in order to determine localization.

Regarding data integrity, we present a method using dynamic data for detection attacks on the cameras in the network. By doing intra-camera correlations as well as inter-camera correlations of spatio-temporal features, we develop a reputation system that is robust to the dynamic environment being observed, yet can detect when attacks occur. Our method determines when a camera has been attacked and is presenting faulty data.

Finally, in addressing data privacy, we look at the social concerns surrounding camera networks in public places and how video data affects privacy. We present a study on what privacy expectations individuals have in a public place and different factors that influence these expectations. Additionally, we look at different technical measures and examine whether they can uphold these privacy expectations.

Above all our hope is that this dissertation will aid in making current camera networks and dynamic scene information more beneficial. Additionally, we hope to inspire others to explore how computer vision can aid in real applications and go beyond the single frame, incorporating multi-view and dynamic information.

Professor Shankar Sastry
Dissertation Committee Chair

To Mindy first because she asked,

To Melissa for the power of three,

And to those two brave souls who raised us all,

And love us unconditionally.

Contents

List of Figures **v**

1 Introduction **1**
1.1 Evolution of Camera Networks 1
1.2 Challenges of Current Camera Networks 6
1.3 Our Contributions to Tackling the Challenges 10

2 Introduction and Background on Camera Network Localization **14**
2.1 Sensor Network Localization and the Importance of Image Data . . . 16
2.2 Related Work on the Use of Image Data in Localizing 18
2.3 Challenges In Using Motion 24
2.4 Contributions of Our Approach 26

3 Localization Using Object Image Tracks **30**
3.1 Problem Formulation 31
3.2 Intra-Camera Track Formation 35
3.2.1 Inter-Camera Track Matching and Correspondence 38
3.3 Experiments 42
3.3.1 Matlab Simulated Network 42
3.3.2 Lab Camera Network with Two Cameras 43
3.3.3 Building Camera Network with Three Cameras 44
3.3.4 Outdoor Camera Mote Network with Two Cameras 46

4 Full External Calibration Using Data Fusion **51**
4.1 Problem Formulation 52
4.2 Overview Radio Inferometry 54
4.3 Fusion-based Localization 56
4.3.1 Linear Method 57
4.3.2 Nonlinear 59
4.4 Experiments using Outdoor Camera Network with Six Cameras . . . 62

5 Localization Discussion **67**
5.1 Incorporating Known Parameters 68
5.2 Toward Data Correlation 70

6 Introduction to Camera Network Attack Detection **72**
6.1 Camera Network Security and the Importance of Image Data 73
6.2 Related Work for Attack Detection 76
6.3 Contributions of Our Approach 78

7 Attack Detection Using Image-Based Reputation **80**
7.1 Problem Formulation 81
7.1.1 Network Setup 81
7.1.2 Types of Attacks 83
7.2 Beta-Reputation Expanded 88
7.3 Camera Methods 93
7.3.1 Static Features 94
7.3.2 Overlapping Motion Features 97
7.3.3 Non-overlapping Motion Features 101
7.4 Experiments 103
7.5 Static Features 103
7.5.1 Overlapping Cameras and Overlapping Motion Features . . . 109
7.5.2 Non-Overlapping Cameras and Non-Overlapping Motion Features 113
7.6 Conclusion 114

8 Introduction to Privacy and Camera Networks **117**
8.1 Related Work for Visual Privacy 120
8.2 Contributions of Our Work 122

9 Subject Study on Privacy Expectations in Public Places **126**
9.1 Study Design 126
9.1.1 Research Questions 127
9.1.2 First Session: Scenario for Subject and Walk 128
9.1.3 First Session: Surveillance Setup 130
9.1.4 Second Session: Image Filtering Techniques 132
9.1.5 Second Session: Follow-up and Survey 135
9.2 Results from Subject Participants 137
9.2.1 Results 137
9.2.2 Visual Privacy Expectations 137
9.2.3 Visual Filters 141
9.2.4 Discussion 146
9.2.5 Privacy Expectations 146
9.2.6 Filters in Relation to Privacy 149

10 Visual Privacy Discussion **154**
10.0.7 Current State of the Technology and Recommendations 155
10.1 Towards a Balance with Visual Privacy 157

11 Conclusion **160**

Bibliography **162**

List of Figures

1.1 An example camera network with each camera labeled as C_i where $i \in 1, 2, ..., 17$ 7
1.2 A example of a camera network where the integrity has been compromised due camera C_5's change in field of view 9

2.1 Localization Example 15
2.2 (Top Row) Images from two wide baseline cameras and it is difficult to tell how the cameras' fields of view are related. (Bottom Row) Images from the same two cameras with SIFT features applied and the matching shown. The matching is incorrect as the static features in each image are not the same. 20
2.3 (Top Row) Images from two cameras. (Bottom Row) Images from the same two cameras with SIFT features applied and the matching shown. The matching is incorrect as the static features in each image are not unique enough to provide unique matches between the images. 21
2.4 Wide-baseline cameras with no moving objects in the scene 22
2.5 Wide-baseline cameras with a moving object in the scene 23

3.1 Track formation: formation of tracks based on object motion in two separate cameras. 34
3.2 Track Matching: (a) shows the tracks in a single camera (b) shows the tracks from another camera over the same time period and(c) shows the correct track matchings between the two cameras that is used in the epipolar constraint. 34
3.3 Candidate Matching: (a) illustrates the possible correspondences between tracks in two cameras; (b) illustrates one matching of overlapping tracks that does not qualify as a candidate match as only 3 tracks match up; and (c) illustrates a good candidate match of overlapping tracks. 40

3.4 Simulated Camera Networks. (a) a 3D perspective view of the network. (b) and overhead view of the network. 43
3.5 Lab camera network with rigid objects 44
3.6 Building camera network. (Top Row) Views from the three cameras without objects in the scene. (Bottom Row) Views from the same three cameras observing people walking. 45
3.7 Position Error: The error in the estimated position, up to scale, from the tracks is given in degrees here. The coordinate frame of the center camera is chosen as the world coordinate frame and all other coordinate frames are aligned to this. 46
3.8 Orientation Error: The error in the estimated orientation from the tracks is given in degrees here. The coordinate frame of the center camera is chosen as the world coordinate frame and all other coordinate frames are aligned to this. 46
3.9 Real Camera Network: (Top Row) Images from the cameras with no moving objects. (Middle Row) Images from on data set with tracks of moving objects shown over time. (Bottom Row): Images from on data set with tracks of moving objects shown over time 47
3.10 Camera Mote Views. (Top Row) View of the courtyard without any moving objects. (Bottom Row) View of the courtyard observing people. 48
3.11 (Top) Image frames from the left and right camera motes, respectively, viewing the scene. (Bottom) The detected foreground objects from the scene. 48
3.12 The tracks of the moving objects in the image planes of the left and right camera motes, respectively, formed by MCMCDA. 49
3.13 (Left) The matching of tracks between the cameras that were used for localization. (Right) The reprojection error measured in pixels for each of the 20 points of the tracks. 50

4.1 The structure of the data fusion localization method. 53
4.2 Two transmitters A and B transmit at the same time at two close frequencies. The interfere signal is observed by receivers C and D. Figure from [48]. 55
4.3 The overlap in fields of view between the cameras nodes based on the object image tracks 58
4.4 An overhead view of the layout of the cameras 63
4.5 Some of the moving foreground objects in the scene as observed from camera 101 64

4.6 (Top) Object image tracks for frames 1 through 500 for cameras which had correspondence in their respective tracks (Bottom) Object image tracks for frames 1 through 500 for cameras that did not have correspondence with the top row cameras, but which had correspondence with each other. 65
4.7 The overlap in fields of view between the cameras nodes based on the object image tracks 66

7.1 Camera Network Setup. 82
7.2 Three different camera setups: (a) a single non-overlapping camera (b) two cameras with overlapping fields of view and (c) two cameras with non-overlapping fields of view, but correlation on motion between them. 84
7.3 Lens Covering and static features: Images from the attack where this camera's lens is obscured by a bag. (Top Row) Subfigures (a) and (b) shows the field of view of the camera before it is obscured by the bag in subfigures (c) and (d). (Bottom Row) Subfigures (e) and (f) show that all the features from the static feature set are detected before the attack and (g) and (h) show the static features in red that should be detected and the actual features that are detected in green which do not correspond to the static feature set. 104
7.4 Lens Covering of a Single Camera:The graph shows the difference between using instantaneous percentages of features seen, an average of the number of features points seen, and using a reputation based on static features. The lens of the camera is covered by a bag at frame 75. The instantaneous percentages are very sensitive to occluding objects hiding features while the average is not sensitive enough to changes in the scene as it takes a while to fall off after the lens has been covered. The reputation system provides a good balance of sensitivity and reactivity. 105
7.5 Moving a Camera Attack. (Top Row) Original views from the fields of view of the cameras. (Bottom Row) Views from the cameras with the third camera being moved. 106
7.6 Moving of a Single Camera:The graph shows the difference between the reputation, based on static features, of the cameras that are not moved and the camera that is moved. Camera 4 if moved at frame 42 so that it's field of view points in a different direction. 106

7.7 Blinding Camera with Laser Attack. (Top Row) shows images from the attack. (Bottom Row) Shows the static features. The red features are the set of robust static features. The green features are the sets of detected features for that image. In the first two images in this row, the robust static features overlap with the detected features, thus no green features appear. In the left two images, the detected features do not overlap with the robust static features. 107
7.8 Lens Blinding of a Single Camera:The graph shows the difference between using instantaneous percentages of features seen, an average of the number of features points seen, and using a reputation based on static features. The lens of the camera is blinded by a laser at frame 84 and after. The instantaneous percentages are very sensitive to occluding objects hiding features while the average is not sensitive enough to changes in the scene as it takes a while to fall off after the lens has been blinded. The reputation system provides a good balance of sensitivity and reactivity. 108
7.9 Original views from the cameras with a moving object in the scene . 111
7.10 Sequence of moving object in a pair of cameras 111
7.11 Replay Attack. (Top Row) The camera where the replay attack happens. The second two frames show the views when the replay attack is occuring. (Bottom Row) Another camera in the network that is trustworthy and has not been attacked. As can be seen, the two cameras should both not see any moving objects if they are trustworthy, but the top camera shows moving objects due to the replay attack. 112
7.12 Replay Attack: These graphs show the reputations of the cameras during a playback loop for the replay attack which begins at frame 50 in camera 1. the graph shown in (a) shows the static feature reputation for each camera while the graph in (b) shows the pairwise reputations. Only the pairwise reputations show any sign of attack while the static reputations for all the cameras look similar. 115
7.13 Nonoverlapping Cameras 116
7.14 Replay Attack on Non-Overlapping Cameras. (Left) The model of the transition time distribution. The bar graph are the results from the training sequences and the gaussian curve is fitted to this data. (Right) The pairwise reputation R_{ij} for when the given replay attack happens on camera C_i. 116

9.1 (Top Row) Frames from the three recorded activities for one subject. (Bottom Row) The same three activities with the camera zoomed in for a closer shot. 131

9.2 (First Row) The head obscuring filter at the bus stop activity. (Second Row) The patch-based blur shown on the stairs activity. (Third Row) the solid silhouette filter shown on the third activity at the stairs. (Fourth Row) The outline filter shown at the bus stop activity. 136
9.3 Activities: (Top) the ANOVA table for privacy results across the 10 activities (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the activity 142
9.4 Places: (Top) the ANOVA table for privacy results across the 10 places (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the place 143
9.5 Identifying Factors: (Top) the ANOVA table for privacy results across the 3 identifying factors (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the identifying factor 144
9.6 Comfort level with the filter on versus off done for each filter independently. The ANOVA table is shown above the box plot each of the subfigures 152
9.7 The filter ratings with the ANOVA table shown above the box plot each of the subfigures 153

Acknowledgments

This dissertation has benefitted tremendously from the wisdom and support of many individuals. The members of my thesis committee are due a special thanks for their mentoring. I am grateful to my advisor, Professor Shankar Sastry, for his support and guidance through both my masters and doctoral projects. I have been very fortunate to have had the opportunity to work with him. I would also like to thank two amazing women, Professor Ruzena Bajcsy and Professor Pamela Samuelson, for their helpful criticism and comments.

I am deeply grateful to my parents, sisters, and extended family, who have been bastions of unwaivering support, the likes of which every scholar should enjoy. Their love and belief in my abilities has helped me to persevere when I encounter obstacles. I consider myself extraordinarily privileged to have such a family.

A very special thanks to Deirdre K. Mulligan whom I have had the great fortune to work with. She has taught me much about the intersection of society and technology and has inspired me to go beyond just the technical details. I am grateful for her mentorship and guidance.

Additionally, I would like to thank Jennifer King. Collaborating with her on privacy research has been an enjoyable experience. I appreciate the discussions on everything from RFID to music and her understanding in some difficult times. I am fortunate to have her as both a colleague and friend.

I wish to thank Dr. Gelareh Taban who was especially supportive during the last

stages of writing this thesis. I am honored to call her my friend and deeply appreciate her kindness, fierce intelligence, and amazing sense of humor.

There are numerous other friends, faculty and staff who have made my graduate experience one that I will cherish forever. Their willingness to listen, give advice, and help me when I needed it most has meant much to me. To Phil Loarie, Maria Jauregui, Dr. Alvaro A. Cardenas, Dr. Vincent Duindam, Dr. Hayley Iben, and many others. Thank you.

Chapter 1

Introduction

I hate cameras. They are so much more sure than I am about everything.
John Steinbeck (1902-1968)

1.1 Evolution of Camera Networks

Cameras have been around in some form or another since approximately 1826 [91], but it is only within the last century that the capability to capture the dynamics of a scene through faster exposure time has come about. Along with this has come the use of camera networks to monitor dynamic scenes. Camera networks consist of some number of cameras which have sensing, communication and processing capabilities. These cameras are used to monitor a designated environment and transmit the visual information to receivers in the network - which can be anything from a central base station with monitors to other cameras in the network - in order to perform some

specified task(s).

In their earliest forms, camera networks were used to transmit images from a remote location to a monitoring station where these images would be displayed. One of the first examples of a camera network was the closed-circuit television (CCTV) system installed by Siemens AG in 1942 in Peenemnde, Germany [22]. Used to observe the V-2 rockets, this CCTV system consisted of a few remote black and white cameras which just transmitted the visual information back to a monitors in the bunker so people could watch the rocket launch.

Technology continued to advance, changing the capabilities of camera networks. In the 1960's, switch boxes were added to camera networks which allowed operators to change between cameras. This enabled operators to switch between multiple camera views on one monitor, but only one camera could be viewed at a time. However, these systems were still relatively simple with black and white images, very poor resolution and coaxial cable in use as part of the system. Individual cameras had to be individually connected to a monitors, which were also black and white.

In the 1970's, multiplexers, video casette recorders(VCRs) and solid state cameras changed the face of camera networks once again. With the use of multiplexers, now the viewing screen could be partitioned to show multiple views on the same monitor. The use of VCRs allowed recording of the images from the cameras and video distribution. This was a major shift from previous camera networks which were mostly used to transmit images, but not record or store them for any significant period of

time. While the first video tape recorder (VTR) was invented in 1951, it was quite expensive, costing $50,000, and difficult to use [29]. The VCR was much less expensive, costing a couple thousand at most, and used removable videotape cassettes containing magnetic tape for recording [3]. Solid state cameras helped improve reliability and the integration of VCRs.

Even with these advances, camera networks were hard to use, expensive, and not very prevalent. VCRs were not reliable for recording and the quality of the recordings was very poor. The combination of low resolution camera images, poor quality video tapes and low tech solutions meant that grainy and unclear images could not be relied on even for conclusive identification purposes. There was no way to do motion detection from the VCR film footage. Additionally, these camera networks were expensive for what you got, in regards to both equipment cost and the installation, and typically only used in limited areas.

Camera networks have continued to change as both the camera technology and the platforms have advanced. With the cameras, there was a switch color images being transmitted instead of black and white, increased frame rates, higher resolution and the change to digital. Additionally, platforms have changed. Digital video recorder and computers have taken over as recording and storage devices making it easier to transmit and store images. Images can be time and date stamped making it easier to review video. In conjunction with this, prices have dropped for these components making camera networks more feasible to implement. Additionally, the switch to

digital and the ease of storage has made image analysis more feasible.

These changes have lead to two major shifts in current camera networks from their predecessors due to the changes in capabilities and lower cost. First, high quality video rate images are now easier to capture, transmit and store reliably. The monitoring of dynamic scenes can be done at a much higher resolution in real-time and reliably stored in case one wants to go back and review footage. The move to digital also increases the ease with which image analysis can be done. High resolution dynamic scenes can be not only monitored, but with more reliable video rate images, it is now possible to do not only single frame analysis, but multi-frame analysis, which takes into account both the spatial and temporal dynamics of the scene. Today's camera networks have more advanced allow for the detection and following of motion.

The second shift is the expanding size and the changing usage of camera networks. The cost of these systems has come down remarkably, allowing the use of many more camera and for many more applications where it once might have been monetarily and technically infeasible. The number of cameras in the network is no longer simply one or two, limited by coaxial cables linked to monitors. Now camera network vary from a few cameras to up to hundreds and hundred linked together and using wireless or wired technology. Camera networks are seeing an increase in use and their applications in more uncontrolled environment and particularly, public spaces, is more common. In the past two decades, particularly after the events of September 11, 2001, the use of camera networks in public spaces for surveillance has taken off, especially in countries

such as the United Kingdom, where the total number of cameras is estimated at around 4,200,000 [51]. In the U.S., Chicago, New York, New Orleans and other cities have deployments of camera networks to monitor areas of interest for undesirable activities[9, 62, 4, 58].

In addition to surveillance, camera networks are seeing use in industrial processes to supervise the processes that take place under dangerous conditions for humans, particularly in the chemical industry. Many cities and have camera networks monitoring traffic to detect congestion and notice accidents. Many of these cameras however, are owned by private companies and transmit data to drivers' GPS systems. For example, the London congestion charge is enforced by cameras positioned at the boundaries of and inside the congestion charge zone, which automatically read the registration plates of cars [5] . If the driver does not pay the charge then a fine will be imposed.

These two shifts with current camera networks lead to many challenges, both from an implementation viewpoint and a social viewpoint. From the implementation viewpoint, there are challenges in how to setup these camera networks so that the video rate data can be used effectively. While it has become easier to capture, transmit, and store this data with high quality, there are many open questions on how the analysis of dynamic scenes can be done in a more automatic fashion to help achieve the goals of the camera network. From the social viewpoint, there are issues on what this change in data collection means in the new settings that camera networks are

being used for. Particularly, how should camera networks that collect high quality video information interact within public settings and how is private data protected. In this thesis, we address some of these current challenges by looking at the aspects of data correlation, data integrity, and data privacy with the modern state of camera networks.

1.2 Challenges of Current Camera Networks

In looking at the challenges that face current camera networks, data correlation, data integrity, and data privacy are some of the aspects that need consideration. We explain what these terms mean in relation to camera networks and their importance in how modern camera networks operate.

Data correlation: For a network of cameras, correlating the data means knowing how image data from each camera relates to image data from the other cameras in the network. More specifically, it is how each camera's field of view and the scene it's seeing is related to the fields of view of the other cameras in the network and what they see at any given time. For example, Figure 1.1 shows that camera C_7 and camera C_5 have an overlap in their field of view.

Understanding this relation provides the groundwork for additional steps that can help achieve goals of the network. Knowing how the image data between cameras is correlated is a useful building block for many tools in current camera networks where image data from multiple cameras must be used in conjunction with one another. For

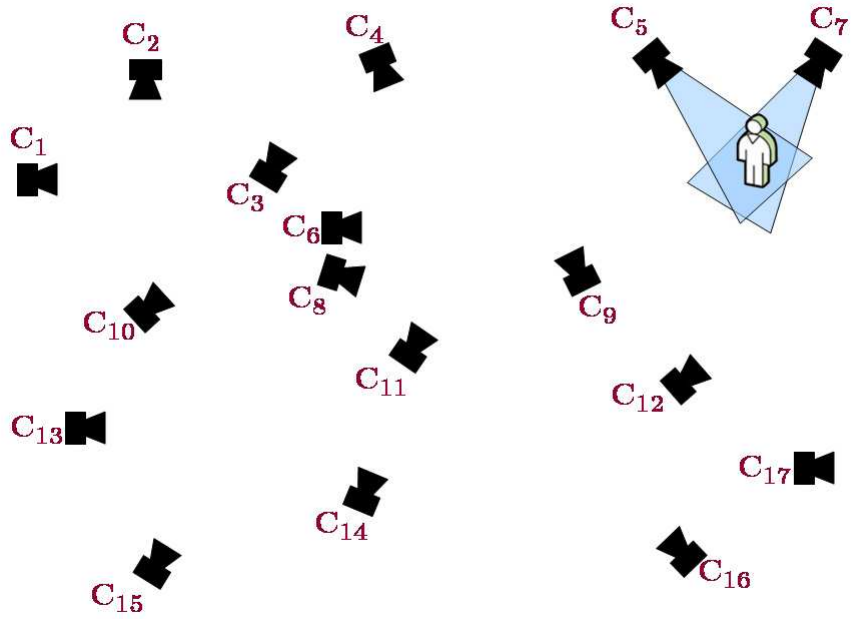


Figure 1.1: An example camera network with each camera labeled as C_i where $i \in 1, 2, ..., 17$

example, if portions of two cameras have identical views, then both do not need to transmit this information, only one needs to be used and redundant data transmission can be eliminated. This can free up bandwidth while still allowing needed information to be delivered. With the growing size of camera networks, bandwidth is an important commodity that must be used efficiently.

If a tracking application is going to be used in a camera network, knowing how each camera's field of view relates to the others in the network is helpful. Once this is known, when one camera is tracking an object a distribution on which cameras should next start tracking that object can be determined instead of looking over all

the cameras in the network to detect the object again.

Data integrity: This is the assurance that the data, the images from the cameras, are unchanged from creation to reception. In order to have integrity, the images the cameras are taking must be of the location(s) they are required to be observing and this image data must be getting sent to designated entities without being altered. Since camera network data is depended upon for differing activities, such as law enforcement and traffic monitoring, the reliability and correctness of that data is critical.

An example of this data integrity getting compromised is demonstrated in Figures 1.1 and 1.2. The setup in Figure 1.1 is what is expected from the network. Cameras C_5 and C_7 are suppose to be monitoring these designated areas and are expected to have an overlap in their fields of view. If the camera network has been compromised by an adversary moving one of the cameras, as shown in Figure 1.2, then the integrity of the data is compromised as the cameras are no longer observing their expected areas. As shown in this figure, camera C_5 has been moved such that its field of view is now covering a completely different area than what is expected. Additionally, there is no overlap between the fields of view of camera C_5 and camera C_7.

As applications of current camera network are increasing, they are being used in public spaces and other uncontrolled areas where the network can be tampered with. Verifying and maintaining data integrity in these situations presents new challenges as the network can be tampered with at any stage in the data flow path by an outside

adversary.

Data privacy: Data privacy is the relationship between collection and dissemination of data, technology, the public expectation of privacy, and the legal issues surrounding them. In relation to camera networks and images, a subset of the topic of data privacy, termed visual privacy, is what must be looked at. Visual privacy can then be defined as the relationship between collection and dissemination of visual information, the public expectation of privacy, and the legal issues surrounding this information.

Current camera networks have created increasing interest in understanding the

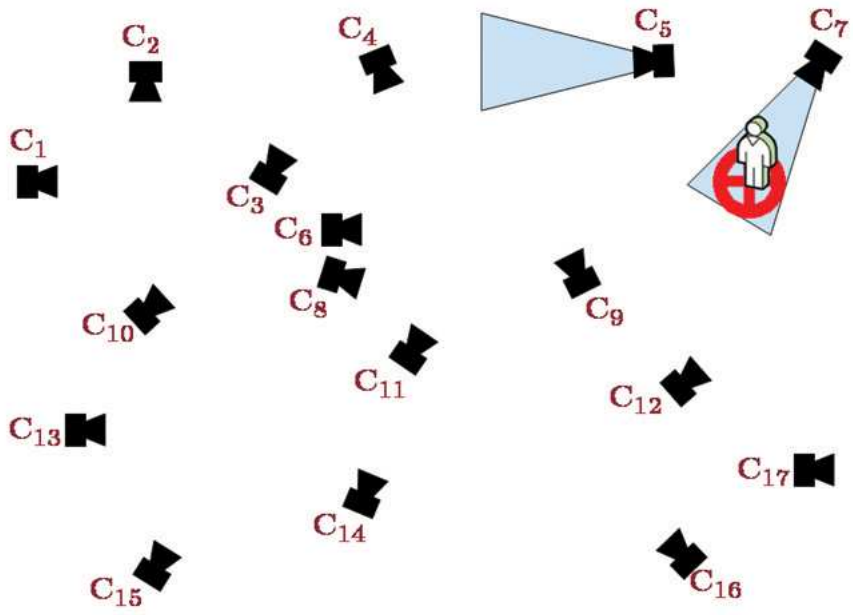


Figure 1.2: A example of a camera network where the integrity has been compromised due camera C_5's change in field of view

advantages and disadvantages of such deployments and how they affect visual privacy. As current camera networks are on a much larger scale than their predecessors, are being used in more public space, and now can more easily collect dynamic visual data, this has increased the interest in visual privacy in regard to this technology. Current camera networks are in public spaces where they can easily capture, transmit, and store any of the actions of the public operating in that space. This visual information can be used for purposes such as surveillance and video analytics.

Maintaining data privacy when using camera networks is important in order to preserve the values society deems important while still gaining benefit from the network. For example, in the U.S. if a camera network is setup to do surveillance in a public space for crime monitoring, but instead the data is used to quell political speech, this does not uphold the first amendment of the U.S. constitution. The right set forth by the first amendment has been deemed as a valuable freedom in society, thus the camera network and its use would be encroaching on this. As camera networks move more and more into the public space, upholding visual privacy becomes increasingly important as more personal data of the public is getting collected.

1.3 Our Contributions to Tackling the Challenges

Given the aspects of data correlation, data integrity, and data privacy, we have structured a set of projects that tackles discrete issues within each of these definitions with regard to current camera networks. With these projects, the goal is to help make

camera networks operate more effectively in their current roles.

In terms of data correlation, we develop a method for localizing cameras in a network that can be applied to a variety camera networks. In particular, it can be applied to wide-baseline networks that are becoming more common and networks where static features are not adequate. Current camera networks use many cameras in varying setups and these cameras cover a much larger physical space than in the past. Thus, determining this correlation for current camera networks is not straightforward as there may be no common visually similarity between them to base correlation on or ambiguous visual similarity in features resulting in multiple correlation possibilities. By taking advantage of the dynamic scenes these current networks can capture, we use the motion of objects from these scenes as a means for determining the localization of the cameras in the network and how each camera's field of view relates to one another. An overview and background on localization is given in Chapter 2. In Chapter 3, we discuss our method using motion from objects in the dynamic scene and present results of this method on different camera network setups. In Chapter 4, we extend this method by combining the motion data, through data fusion, with radio information. We conclude with a discussion of the localization in Chapter 5.

In addressing the data integrity element, we look at how to detect intrusion attacks on the cameras themselves, where the image data is being obtained. As current camera networks are being used in more uncontrolled, public, environments where they can be tampered with, it is important to determine when image data from the

cameras is reliable or not. Methods for protecting data while in transmission and determining if it has been tampered with, such as encryption and watermarking, already exist. Additionally, on the storage end, there are methods, for example access keys, that can be used for protecting the data. However, little has been done in looking at how to determine if the cameras themselves have been tampered with and determining if they are sending faulty data.

We present a reputation system based on spatio-temporal image correlation data from the dynamic scenes to determine if a camera has been tampered with. By using varying types of features to build a reputation, we show what kinds of attacks can be detected in varying types of camera setups. A background on reputation and attack detection in is given in Chapter 6. In Chapter 7, we present our reputation system for detecting attacks on cameras in the network and the varying visual and motion features that are used. Additionally, results of this method on different camera network setups is given.

To deal with the visual privacy element, we look at the expectations of privacy in terms of visual data in surveillance in public spaces. Visual privacy is an open and rather undefined area and there are varying policies, yet very little has been done in the way of looking at public expectations in the spaces where surveillance is happening. Through the subject study, we give an insight into what these expectations are. To try and tease out public expectations on camera networks in public spaces, we present a case study of subjects being surveilled and their reactions to the

surveillance. I present technical solutions based on image analysis as privacy measures and see whether these methods uphold their expectations. An overview on different camera network policies laws related to image data is given in Chapter 8. The case study on public expectations of privacy and proposed technical solutions for camera networks is presented in Chapter 9. We conclude with a discussion of results from both and future directions for maintaining privacy in camera networks in Chapter 10.

Chapter 2

Introduction and Background on Camera Network Localization

We're not lost. We're locationally challenged.

John M. Ford (1957-2006)

Localization plays an important part in any network setup as it is a building block for many applications from data fusion to security protocols. Localization in regards to networks of sensors means determining where the sensors are in relation to one another and how their sensing fields are related. In terms of camera networks, this means knowing the positions of the cameras relative to one another as well as the orientation of each camera's field of view. An example of this is shown in Figure 2.1. The fields of view of cameras C_5 and C_7 overlap and are related by the extrinsic parameters for orientation, R_{57}, and position, $\lambda_{57}T_{57}$.

In camera networks, knowing how the fields of view of each camera relate to

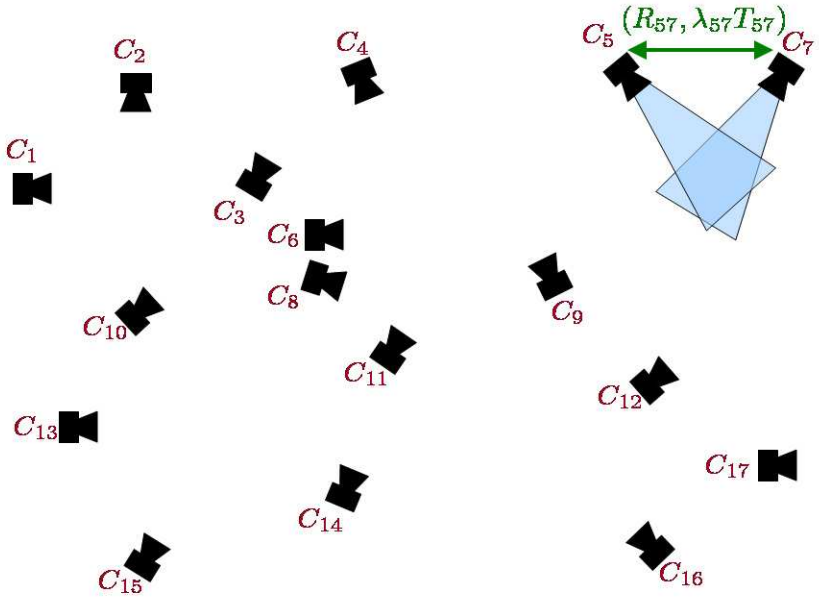


Figure 2.1: Localization Example

the others in the network is the building block for many tools in current camera networks where image data from multiple cameras must be used in conjunction with one another. In this chapter, we provide a brief an overview of localization, how it has been done in traditional low-bandwidth sensor networks, and how it is developing for high-bandwidth camera networks. We discuss the importance of using the motion information current camera networks captures from the dynamic scenes and how our approach takes advantage of this for localization.

2.1 Sensor Network Localization and the Importance of Image Data

Camera networks can be thought of as high-bandwidth networks of sensors as they are made up of spatially distributed autonomous devices with sensing capabilities that cooperatively monitor an environment. The term *sensor network* traditionally refers to low-bandwidth sensors such as temperature, sound, vibration, pressure, and radio sensors. For the purposes of this section, we will differentiate between these types of sensor networks and camera networks by respectively referring to them as low-bandwidth and high-bandwidth networks.

Localizing networks of sensors is not a new field and is well examined in the low-bandwidth sensor network community. In this community, localization is an important aspect for effective use of the network. There are differing automatic methods that have developed for localization of low-bandwidth networks. Many current automatic low-bandwidth sensor networks are localized using acoustic information and radio frequency intensities. These are active methods in which acoustic or radio signals are sent out and the received signal strength indicators, time of arrival, time difference of arrival, or angle of arrival are used to determine the localization of the sensors [42]. Using these features, the position of the sensors in 2D and sometimes 3D space can be determined.

As low-bandwidth sensors are often not directional, knowing the position alone

allows higher level methods that need localization, such as data fusion, multi-hop routing, and security measure such as key-generation, to run on these networks. However, in translating these low-bandwidth methods over to camera networks, these methods do not provide all the localization parameters necessary for camera networks to perform higher level tasks, such as image data fusion or camera handoff, as there is no information on field of view orientation of the cameras. As cameras are directional sensors, the position alone will not give the all the relevant data in order to determine how to interpret the image data in these higher level tasks. Thus, even if radio sensor were attached to cameras in a network, the localization methods used in low-bandwidth sensor networks would not provide enough information to localize the cameras.

In order to perform tasks with visual information, such as tracking across multiple cameras, it must be known not only where the cameras are positioned in the 3D space, but also how the cameras' fields of view are oriented and how they overlap with other cameras' fields of view. Since this information is specific to the visual input gathered by the cameras, it is logical to develop a localization method based on using the images from the cameras. In our work, we look at how to use visual information, specifically from dynamic scenes, in order to localize the cameras in the network.

2.2 Related Work on the Use of Image Data in Localizing

In using visual information for localization, computer vision researchers have made tremendous progress in doing automatic image alignment and determining the orientation, R, and position, T, up to scale, in multi-view settings. Using visual information, geometric constraints have been well established that allow for one image to be warped into another. Thus, it can be determined how one image is related to another. Solutions to satisfy these constraints rely on detecting common appearance-based features between both images. This is normally done in three steps: detecting the features, encoding them using feature descriptors, and finally matching them against each other. Many affine covariant region detectors are available, namely, the MSER detector [14], the Salient region detector [86], the Harris-Affine detector, the Hessian-Affine detector [55], the Intensity extrema based detector (IBR), and the Edge based detector (EBR) [90]. There are also quite a number of feature descriptors such as the Scale Invariant Feature Transform (SIFT) [45], Gradient Location and Orientation Histogram (GLOH), Shape Context [10], Moments [30], Steerable filters [25] and cross correlation of pixels. Important attributes are invariance to scale, rotation, transformation or changes in illumination. A good overview over both detectors and descriptors can be found in [57, 56]. Using these types of methods work well when the fields of view of the cameras in a network are related to each other by a

small baseline. Given this, automatic appearance-based features can be detected. For example, in [47], the authors localize a network of cameras by assuming there is a common set of visual features seen by each set of three cameras that can be used for external calibration.

These previous methods rely on cameras sharing similar static feature with similar appearance. However, this is often not the case in many camera networks, as they can be wide baseline and even if they see the same objects in the scene the image features of that object look different in each camera and do not lend well to visual appearance correspondence. For instance, in Figure 2.2, these wide-baseline cameras are viewing the same scene, yet if an automatic feature detection method is used, there is an incorrect matching of features as the images do not contain the same static features. Additionally, there may be an ambiguity in visual features if they are shared between cameras. An example of this is shown in Figure 2.3. The corner points from all the tiles on the floor are the feature points automatically detected and their correspondence between images is ambiguous. Thus, there are multiple hypotheses that result for how the two cameras' fields of view are related.

These examples demonstrate some of the challenges in automatically localizing camera networks. In camera networks where there are wide baseline cameras the issue that the common scene observed by two cameras will not necessarily look the same between the cameras and this is a challenge. There will be no common appearance cues and thus, we cannot use appearance-based features. Traditional image feature

Figure 2.2: (Top Row) Images from two wide baseline cameras and it is difficult to tell how the cameras' fields of view are related. (Bottom Row) Images from the same two cameras with SIFT features applied and the matching shown. The matching is incorrect as the static features in each image are not the same.

detection methods rely quite a bit on visual appearance and therefore will not work given this case as are no similar static features between cameras and if there are, they do not have the similar appearance. An additional challenge is that the cameras may have some common appearance cues, but they are not unique enough to provide good correspondence information to do localization. In light of these challenges, how do we then find features which can be correlated across cameras?

One approach is to go back to manual intervention and manually measure the pose or use a calibration pattern to do a semi-automatic localization. In [83], the authors take this approach and use a point light source to calibrate cameras. Additionally the

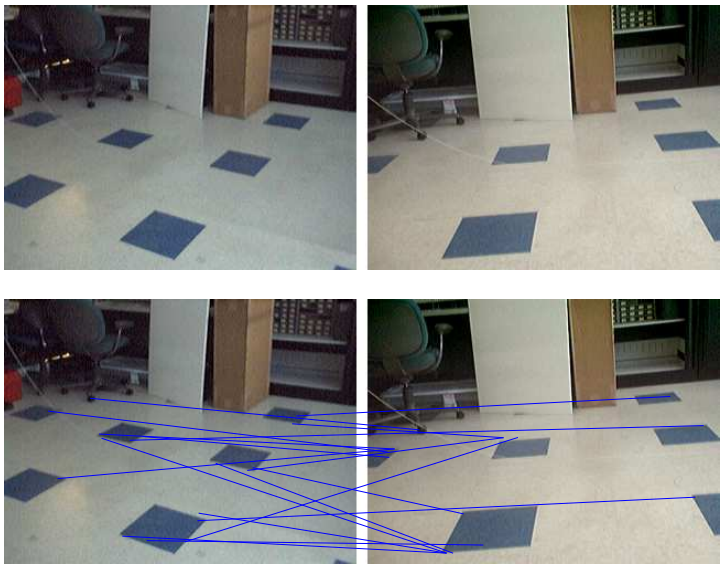


Figure 2.3: (Top Row) Images from two cameras. (Bottom Row) Images from the same two cameras with SIFT features applied and the matching shown. The matching is incorrect as the static features in each image are not unique enough to provide unique matches between the images.

EasyCal Calibration Toolbox [2] uses a point light source to calibrate a network of cameras. This is a semi-automatic method where the light source is manually moved around in observation space of the cameras and creates common features points shared by the cameras. This method requires conditions in which the light source can actually be seen by the cameras as well as requiring time consuming manual intervention.

Manual methods are very tedious and time consuming and sometimes requires special environmental conditions that may not be present in camera network setups. In looking at how to create a fully automatic method that does not rely on appearance feature similarity, there has been a movement towards looking at moving objects in

the scene. When looking at dynamic scenes, a human can infer how the views from cameras are related by looking at the consistency of the motion of objects between the cameras. Looking at the scene with no moving objects in it, as shown in Figure 2.4, it is unclear how the cameras are oriented with respect to one another. However, when a moving object enters the scene, as shown in Figure 2.5, it becomes more clear how the cameras are positioned relative to one another.

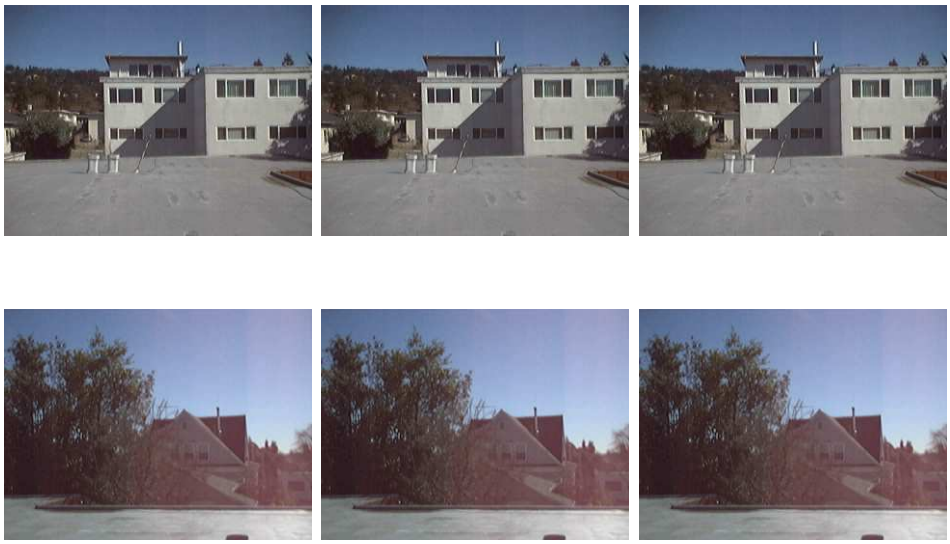


Figure 2.4: Wide-baseline cameras with no moving objects in the scene

In [26], the authors use a statistical method to find the localization parameters of the cameras. A known object is tracked over time and the image position at each time is recorded and used as a means to solve for the localization parameters. However, it is assumed that the object is known ahead of time and the height, the Z coordinate of the position vector, of the cameras as well as two of their orientation parameters,

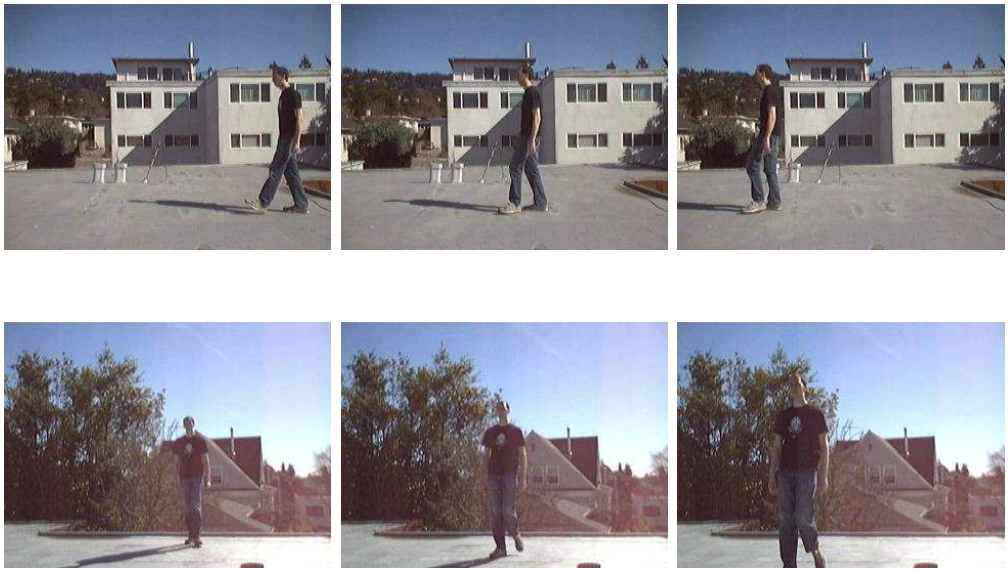


Figure 2.5: Wide-baseline cameras with a moving object in the scene

are already known. Only the X-Y portion of the position vector and the orientation around the principal axis of the camera are solved for. There is still an assumption of one object in the scene making it easy to track and identify.

In [74, 20], the authors assume a common global ground plane between all cameras. Within a single camera, detected objects are fit to a local ground plane and then using homography constraints, these local ground planes are matched to a global ground plane. In these works, how the camera is oriented to a local ground plane is already known and the objects in time are used to provide the constraints for the homographies and determine which camera field of view relates to which others. In [11, 43], this ground plane concept is extended a bit, where the relation of the local ground plane to the camera is not known, but solved for based on known objects and

then these local ground planes are aligned to a global ground plane based on homographies. While these methods work well in certain settings, they are not generalizable as it is not always the case that a common global ground plane exists and trying to fit one to the data from the cameras can lead in wrong localization parameters.

As with these previous work, we feel that using moving objects in the scene is key to localizing cameras in a network, particularly in wide-baseline settings. Differing from these authors, we take advantage of the actual motion of the objects in both the spatial and the temporal space in order to create a general method that is independent of the scene structure and does not rely upon known objects or a fixed number of objects. The goal is that this general method can be adapted to take into account any knowns, such as number of objects or scene structures like ground planes, if these things are known. However, this method can also be used when there are none of these givens and the structure of the scene and moving objects are unknown to estimate the localization of the camera network. We use the dynamic scenes that current camera networks can capture, and use the motion of objects in the scene as a cue to help with localization. By looking at the consistency of motion between cameras, rather than visually similar features, we demonstrate how localization of cameras can be done.

2.3 Challenges In Using Motion

In using moving objects in the scene and exploiting the fact that current camera networks are able to capture and dynamic scene information, there exist challenges in

how to use this data. First, since moving objects may have different appearance cues, other aspects of the moving objects must be exploited and used for localization. We look at the spatio-temporal *motion* of the object in the image plane of each camera as the feature(s) to use for correspondence between cameras in the geometric space via the epipolar constraint.

This leads to the next challenge of how to represent the motion of objects in the image plane? Should optical flow be used and if so, how are flow vectors compared? How should motion be represented so it can be used effectively as a feature for localization? Since flow vectors are based off of appearance and appearance varies between cameras, we chose not to use optical flow vectors as a measure of motion. Instead, we abstract each moving object to a blob and then use the centroid of that blob over time as a measure of motion . By associating the centroid of an object over time, we build an object image track in each camera that observes the object and use this as the motion feature. While the centroid of an object in one camera will not map to exactly to the same 3D point of the centroid of the object in another camera, we show that this does not introduce significant error. We also show that given some more knowledge about the network setup, a point other than the centroid, can be chosen on the objects in the image such that these image points do map to the same 3D point.

The final challenge is determining how to correlate the motion track features between cameras. Should the shape of the track be used? Should the distance between

points be used? Appearance cues cannot be relied upon and this must be avoided, so instead we make use of geometric constraints that must be satisfied if two cameras are related and use the points of the tracks as possible correspondences. Since the motion tracks have already been built, this limits the possible correspondences and also additional filtering is done to narrow the possible correspondences by exploiting the time synchronization of cameras in a network.

2.4 Contributions of Our Approach

If we are to automatically localize the cameras in a general network setup using moving objects in the scene, we need to take full advantage of the commonalities in camera network setup parameters, appearance cues *within*each camera, and the video quality data. We break the problem into a two step process. The first step is an intra-camera step where a statistical approach is taken to determine the motion of objects and build the motion tracks in each image plane. Each camera is treated independently at this stage and the static background appearance and video rate data is used to aid in picking out the moving objects to build the motion tracks. The second step is an inter-camera step where geometric constraints are used to determine the correspondence of motion tracks between cameras. Taking advantage of known internal parameters of the cameras and synchronization, or time stamping on the images, helps in the process.

Our approach stands in contrast to past work that recovers camera position, T, up

to scale, and orientation, R, in a mulit-view setting. Previous works are based known objects in a scene, known scene geometry with a common ground plane, and known shared appearance features between cameras. Our method is quite general and can be applied in a network with unknown scene geometry, camera layout, and objects, as long as the moving objects can be detected and isolated. That we can get an estimate of localization based on motion alone without scene knowledge and appearance cues, is a strong testament to the importance of these cues and a data-driven approach.

One key idea in our method is to not focus on particular object or motions, but just detect all moving objects in the scene and let the statistical data association method filter out noisy movements. Those moving objects with a small motion, such as tree swaying in the wind, will not have a path built by data association. Therefore, small background motions can be eliminated.

A second key idea in this work is to not focus on the detailed motion of the moving object, but instead, treat the object as a whole entity and look at the overall motion. While the moving objects in the scene may be rigid or non-rigid, no differentiation between the two is done. Each object is treated as a whole and the overall motion is looked at, not the specifics of the motion such as might be given by flow vectors. This allows an overall motion track of the object in the image plane to be created and to use as a feature. While the non-rigid motions may introduce some noise into the measurement, the data association method used to build the motion tracks has an adjustment step that does a best fit and can eliminate some of the noise.

A third key idea in this method is to break the problem into an intra-camera step and an inter-camera step. This two step approach takes advantage of consistencies within a single camera that do not exist between cameras. The fast frame rate and consistency in visual motion that exist in a single camera between frames is exploited.

Our contributions with this approach of using motion consistency of objects in combination with geometric constraints of projection to provide for the automatic localization of the cameras in a network as follows:

- Breaking the problem into intra-camera and inter-camera steps
- Using moving objects and their spatial and temporal aspects as features within each camera (Intra-Camera Step)
  - Treating moving objects as a whole
  - Building tracks using the moving objects within the image plane of each camera
- Using the tracks from each camera as features to compare between cameras (Inter-Camera Step)
  - Relation of tracks to geometric projection

Additionally, we show how our method using moving objects for localization can be combined with information from radio sensors for complete extrinsic calibration of the cameras in the network. We present a fusion-based localization method which

combines the visual data with the radio data from radio inferometry in order to solve for the position, R, and the orientation, T, of each camera and the scale factor, λ.

Since no previously existing datasets are appropriate to test our localization method on, as either they have no motion or do not have a ground truth to compare results against, we use new data sets based on both simulated and real camera networks of varying size. In testing our visual localization method we created a simulated camera network environment in matlab and then used four different real camera network setups ranging in size from 2 to 6 cameras. An indoor lab setup consisting of two cameras, an indoor building setup consisting of 3 cameras, an outdoor network consisting of two cameras, and an outdoor setup consisting of 6 cameras are all used. To test our extension of the visual localization to the fusion-based localization that uses radio information as well, we use an outdoor setup that consists of 6 cameras and 6 radio sensors.

Chapter 3

Localization Using Object Image Tracks

We will be known forever by the tracks we leave.

American Indian Proverb, Dakota

To localize the cameras in the network we use the moving objects in the scene in order to deal with the previously mentioned complexities that appearance-based features create with modern camera network setups. In each camera, we build tracks of the moving objects, as they appear in that camera's image plane, which we call *object image tracks*. These object image tracks are then used as features for determining localization. In this chapter, we describe our method of localization using object image tracks, originally presented in [53]. We present the necessary assumptions on the network in order to use this method and describe both the inter-camera and intra-camera steps used in determining the orientation and translation, up to

scale, of pairs of cameras.

3.1 Problem Formulation

For this method of localization, the assumptions on the inputs and outputs are:

- **Input:** synchronized video sequences from the N fixed cameras in the network which are at unknown positions and orientations and known internal calibration parameters for all cameras
- **Output:** The orientation, $R \in SO^{3\times3}$ and the position, up to scale factor, $T \in \mathbb{R}^3$ for all N cameras.

No prior assumptions on where the N cameras are placed are made except that there must be some field of view overlap between pairs of cameras if the orientation, up to scale, and position of those cameras are to be recovered. Further, no assumptions on the scene structure are made. For example, the cameras may be wide baseline and, if there exists a common ground plane, no prior knowledge of how each camera's coordinate frame is related to that ground is known. No prior correspondences of features between cameras is known, nor are any assumptions that the same static scene features must appear in multiple cameras.

With this approach the goal is to find correspondences between cameras given that the baseline and photometric characteristics can vary considerably between images from different cameras. Thus, one cannot necessarily use brightness or proximity

constraints and traditional methods of features correspondence, such as SIFT features [45] or Salient Region [87] will not necessarily work for localization. However, by observing moving objects in the scene, this information can be used in order to localize the cameras.

Our localization method consists of two main steps in order to use individual raw video streams to localize the cameras: an intra-camera step and an inter-camera step. The intra-camera step, which is called track formation, involves exploiting similarities of objects between frames for each camera separately. The inter-camera step, which is called track matching, involves using the object image tracks from each camera as features to compare against object image tracks from other cameras. Without loss of generality, the following variables are defined as:

N: the number of cameras in the network
C_i: the ith camera in the network where $i \in 1, 2, ..., N$
P: the set of moving objects in the scene
p_i^t: where $p_i^t \subseteq P$ are the subset of objects C_i observes at time t
Θ_i: the set of object image tracks for camera C_i
R_{ij}: the relative orientation of C_i with respect to C_j
T_{ij}: the relative position, up to scale, of C_i with respect to C_j

An overview of the inter-camera and intra-camera steps are as follows:

1. **Track Formation:** Find moving objects in each camera's field of view and based on correspondences between frames, build tracks of those objects within the image plane as shown in figure 3.1.

   For this step, each camera, C_i, observes some set of objects, p_i^t, at each time t. No assumptions are made on what objects are seen by which cameras nor what

type of objects can be seen. For example, humans, cars, and dogs could all be types of moving objects seen by the cameras.

Additionally, it is assumed that the frame rate of the cameras are fast enough to pick up the motion of objects moving in the scene. Within a single camera, C_i, moving objects, p_i^t, are found and tracks of these objects within the image plane, Θ_i, are built based on multi-target tracking. This is discussed in more detail in section 3.2.

2. **Track Matching:** The image tracks, Θ_i, from each camera are used as features to compare against image tracks from another cameras in order to determine the relative (R_{ij}, T_{ij}) of each camera. This comparison is done in a pair wise manner and is based off of correspondences and the properties of the essential matrix. This is illustrated in Figure 3.3.1.

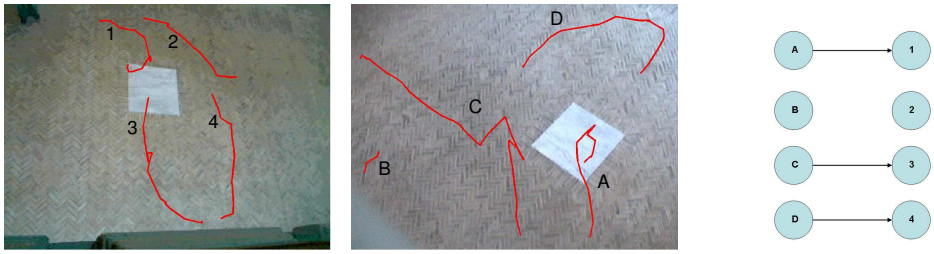
   Multiple view geometry is well studied. For two views there exist geometric constraints that relate the corresponding points in the 3D camera geometry, which come about in the epipolar constraint. Since the the internal parameters of all cameras are known, the focus here is on the essential matrix, based off the epipolar constraint. A detailed description of the epipolar constraint can be found in [46].

   Once object image tracks in each camera have been recovered, these tracks are used as spatio-temporal features. These spatio-temporal features are compared

between each pair of cameras to get a the relative orientation and position, (R_{ij}, T_{ij}), based on the essential matrix. This is discussed in more detail in section 3.2.1

(a) Camera 1 **(b)**Camera 2

Figure 3.1: Track formation: formation of tracks based on object motion in two separate cameras.



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

Figure 3.2: Track Matching: (a) shows the tracks in a single camera (b) shows the tracks from another camera over the same time period and(c) shows the correct track matchings between the two cameras that is used in the epipolar constraint.

3.2 Intra-Camera Track Formation

The intra-camera step is done for each camera C_i independently in order to form the object image track set Θ_i. Filtering the video data in order to find the moving objects in each camera's field of view is the necessary first step. An adaptive background subtraction technique is applied in order to segment moving objects. We use the method proposed by [99], which is an adaptive background subtraction method, in order to do this segmentation. Items that are determined as foreground objects are considered to be possible moving objects in the scene.

After the background segmentation is run, the remaining foreground objects are further filtered. Bounding boxes are formed around each foreground object. If a bounding box is too close to the boundary of the image, based on a threshold of q pixels, this object is filtered out and not counted in the final set, p_i, of moving objects for C_i for that frame. This is done to help make the track formation more robust as only the centroids of fully seen objects will be used. Only the foreground objects that lie completely within the image are treated as the true moving objects. Thus, for each camera C_i we get some set of moving objects p_i^t at each time instance t.

For each moving object in the frame of a camera, the centroid of that object is computed. This leads to a set of centroids, M_i^t, for each frame of camera C_i . The centroids of the objects are then used as the measurements for the multi-target tracking in order to build object image tracks.

In recent years, *multi-target tracking* has received a considerable amount of atten-

tion in the computer vision community because the task of tracking multiple objects in video sequences is an important step towards understanding dynamic scenes. The essence of the multi-target tracking problem is to find a track of each object from the noisy measurements. If the sequence of measurements associated with each object is known, multi-target tracking reduces to a set of state estimation problems, for which many efficient algorithms are available. Unfortunately, the association between measurements and objects is unknown. The *data association* problem is to work out which measurements were generated by which objects; more precisely, we require a partition of measurements such that each element of a partition is a collection of measurements generated by a single object or clutter [81]. Due to this data association problem, the complexity of the posterior distribution of the states of objects grows exponentially as time progresses. It is well-known that the data association problem is NP-hard [16, 72], so we do not expect to find efficient, exact algorithms for solving this problem. The problem gets more challenging with video sequences due to the nonlinear camera projection, occlusions, and varying appearances, to name a few.

Since cameras are not calibrated, we cannot use the 3D model-based tracking approaches such as [34, 21]. However, we can still track moving objects on a 2D image plane. In addition, the computational complexity of the model-based approach, *e.g.*, [21], is not desirable for our rapid autonomous calibration task.

In order to handle highly nonlinear and non-Gaussian dynamics and observations, a number of methods based on particle filters has been recently developed to track

multiple objects in video [34, 66, 38]. Although particle filters are highly effective in single-target tracking, it is reported that they provide poor performance in multi-target tracking [38]. It is because a fixed number of particles is insufficient to represent the posterior distribution with the exponentially increasing complexity (due to the data association problem). As shown in [38, 98], an efficient alternative is to use Markov chain Monte Carlo (MCMC) to handle the data association problem in multi-target tracking.

For our problem, there is an additional complexity. We do not assume the number of objects is known. A *single-scan* approach, which updates the posterior based only on the current scan of measurements, can be used to track an unknown number of targets with the help of trans-dimensional MCMC [98, 38] or a detection algorithm [66]. But a single-scan approach cannot maintain tracks over long periods because it cannot revisit previous, possibly incorrect, association decisions in the light of new evidence. This issue can be addressed by using a *multi-scan* approach, which updates the posterior based on both current and past scans of measurements. The well-known *multiple hypothesis tracking* (MHT) [18, 75] is a multi-scan tracker, however, it is not widely used due to its high computational complexity.

A newly developed algorithm, called Markov chain Monte Carlo data association (MCMCDA), provides a computationally desirable alternative to MHT [64]. The simulation study in [64] showed that MCMCDA was computationally efficient compared to MHT with heuristics (*i.e.*, pruning, gating, clustering, N-scan-back logic

and k-best hypotheses). In this paper, we use the online version of MCMCDA to track multiple objects in a 2D image plane. Due to the page limitation, we omit the description of the algorithm in this paper and refer interested readers to [65] or [64].

3.2.1 Inter-Camera Track Matching and Correspondence

Once Θ_i has been determined for each camera, these object image tracks are then treated as spatio-temporal features. The inter-camera step looks at the correspondence between these features for each pair of cameras.

For a given time period $t_0 : t_n$:

(C_i, C_j):pair of cameras where $i \neq j$
$\theta_i^m \in \Theta_i$: is a specific track in camera C_i where $m \in \{1, ..., |\Theta_i|\}$
$t_s(\theta_i^m)$: starting time of a track $\theta_i^m \in \Theta_i$
$t_e(\theta_i^m)$: ending time of a track $\theta_i^m \in \Theta_i$

where $t_0 \leq t_s(\theta_i) < t_e(\theta_i) \leq t_n, \forall \theta_i \in \Theta_i$.

In doing pair-wise correspondence with object images tracks, between a pair of cameras (C_i, C_j), we see use (Θ_i, Θ_j) for possible correspondences. It is important to note that it is possible just to use the centroids of the moving objects from all the frames alone, without forming tracks, as features and do point correspondences using these points. However, forming the object image tracks by using data association is much more beneficial. By using the intra-camera data association step, the space of possible correspondences is cut down as multiple points in one track are limited to corresponding to multiple points in another single track only, not separate points

from multiple tracks. For example, looking at the pair (C_i, C_j), if the first point on the track θ_i^1 corresponds to a point on θ_j^1, then then another point θ_i^1 cannot correspond to a different track, such as θ_j^2. Additionally, using the track constrains the correspondence space further based on timing data from the on the tracks. Only tracks which have an overlap in time can correspond. Using the object image tracks from the intra-camera data association, thus greatly reduces computation time and further constrains the (R, T) of the cameras, leading us to a more accurate solution.

In doing pair-wise correspondence with object images tracks, the first step is to determine if there are enough tracks that overlap in time between the camera pair (C_i, C_j). A pair of tracks $(theta_i^m, theta_j^n)$ between (C_i, C_j) overlap if there is a sufficient intersection in the sets of track times:

$$|t_s(\theta_i^m), .., t_e(\theta_i^m) \bigcap t_s(\theta_j^n), ..., t_e(\theta_j^n)| \geq 8 \tag{3.1}$$

Define $\Psi : (\Theta_i, \Theta_j)$ to be the process which finds all pairs of the possible overlapping tracks and Φ to be the results set of overlapping pairs, with $\phi_m \in \Phi$ to be a single pair. Once all pairs of overlapping tracks have been found, the first pair is used to find an essential matrix. Using this essential matrix, defined as E_m, the reprojection error for the remaining possible corresponding track pairs is calculated. Those pairs with a reprojection error under a given threshold, g, are deemed as good correspondences for E_m. Define $\Omega(E_m)$ to be the process which finds the set,Γ, of all the good

corresponding pairs, γ_σ, where $\sigma \in 1, 2, .., \mu$ and the total number of good correspondences for the E_m is μ. This can be thought of a pseudo-RANSAC process using tracks instead the traditional points. An example of this candidate matching can be seen in Figure 3.3.

The E_m corresponding to the γ_σ with the largest μ is deemed to be the essential matrix that best represents how the fields of view of C_i andC_j are related to each other. It is then a well known step to recover the extrinsic parameters, R_{ij} andT_{ij} from the essential matrix [46].

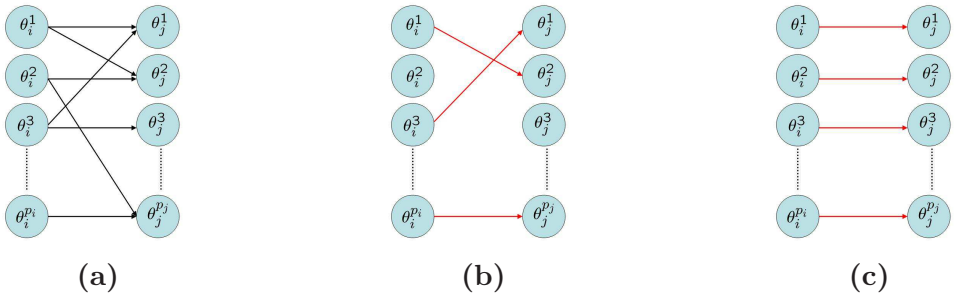


Figure 3.3: Candidate Matching: (a) illustrates the possible correspondences between tracks in two cameras; (b) illustrates one matching of overlapping tracks that does not qualify as a candidate match as only 3 tracks match up; and (c) illustrates a good candidate match of overlapping tracks.

The pseudo code for the whole process is as shown in Figure 3.2.1:

A natural question to ask is whether tracks based on the centroid of an object will actually correspond to the same points in space when dealing with the epipolar constraint. If the segmentation is perfect and the objects are spheres, then the cen-

Algorithm 3.2.1: LOCALIZATION($rawvideo$)

for $t \leftarrow 1$ **to** t_{max}

for eachC_i, build object tracks

for each pair (C_i, C_j)

  do find candidate matches using $\Psi(C_i, C_j)$

$|\hat{\Gamma}| = NULL$

$\hat{E_{ij}} = NULL$

if $|\Phi| \geq 2$

$$\textbf{do} \begin{cases} \forall \phi \in \Phi \\ \text{Find } E \\ \text{find good matches using } \Omega(E) \quad \textbf{return } (\text{all}E_{ij}) \\ \textbf{if } |\Gamma| > |\hat{\Gamma}| \hat{E_{ij}} = E \\ |\hat{\Gamma}| = |\Gamma| \end{cases}$$

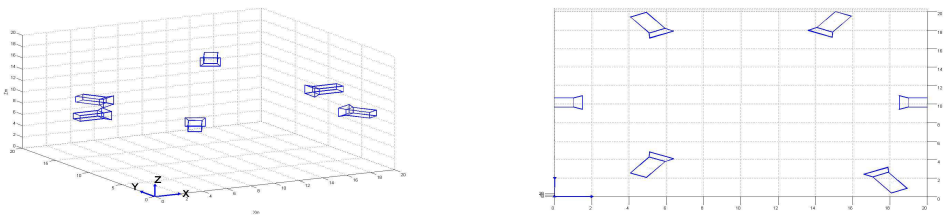
troids of the segmented objects do in fact correspond to the same points in space. Yet most objects do not have this nice property nor can be segmented out perfectly in the image such as people, cars, dogs, and other types of moving objects often in scenes. However, as an upper bound on the error, we know that the true centroid of the object will be inside the convex hull of the silhouette segmented out. Our results show that there the reprojective error in the results is very small and the estimates for all (R, T) are within 8 degrees, in both position and orientation, of the ground truth results.

3.3 Experiments

We tested our method of localization using object image tracks in four different setups.

3.3.1 Matlab Simulated Network

In this setup, we simulated objects moving through a scene observed by a simulated camera network in MATLAB. Seven camera views were simulated and different size cubes were used as the moving objects in the environment. Perspective projection and triangular fields of view were used for the simulated cameras. All measurements had gaussian noise of $N(0, 1)$ added to them. The camera setup can be seen in Figures 3.4a and 3.4b.



(a) (b)

Figure 3.4: Simulated Camera Networks. (a) a 3D perspective view of the network. (b) and overhead view of the network.

Using the ground truth to compare measurement for the corners of the cubes, the reprojection error in the estimated measurements for the corners was off by an average of 1 pixel.

3.3.2 Lab Camera Network with Two Cameras

In this setup, two Logitech QuickCam 4000 webcams were setup in a lab room observing an open area. The cameras captured 5 frames per second with an image size of 320 x 240. Three minutes of video were taken in total. Rigid objects were moved through the area and observed by the cameras. This can be seen in Figure 3.5. Since we already had the knowledge that the camera network was looking at a common ground plane, we solved for the homography between the cameras, instead of the essential matrix [46]. This setup let us test the algorithm on rigid objects of moving in a plane.

We obtained ground truth external camera parameters for this setup by turning

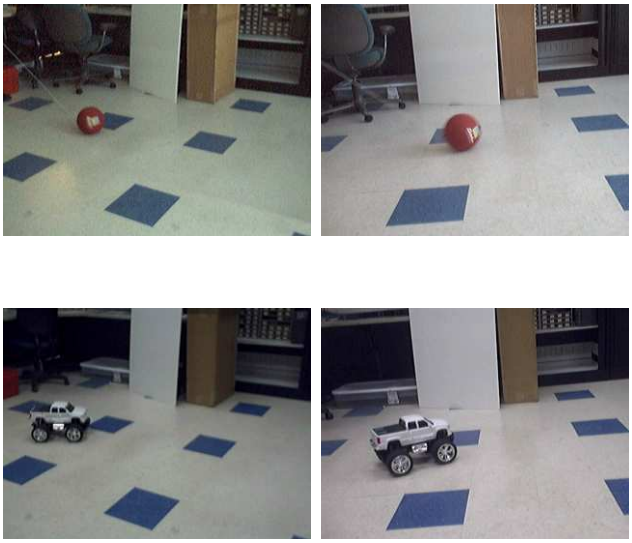


Figure 3.5: Lab camera network with rigid objects

off the lights and using a point light source to obtain consistent feature points. The average reprojection error using the results from the homography was 1.1 pixel.

3.3.3 Building Camera Network with Three Cameras

In this setup two Logitech Orbit MP cameras along with a Logitech QuickCam 4000 camera were setup on the second floor of an atrium looking down on the floor below. The cameras observed people walking through the space. This can be seen in Figure 3.6. This setup let us test the algorithm on articulated objects of different sizes and speeds moving through a space.

Tracks of people walking on the first floor were used to localize the cameras. Three different sequences of 60 seconds of video footage were used. As we had

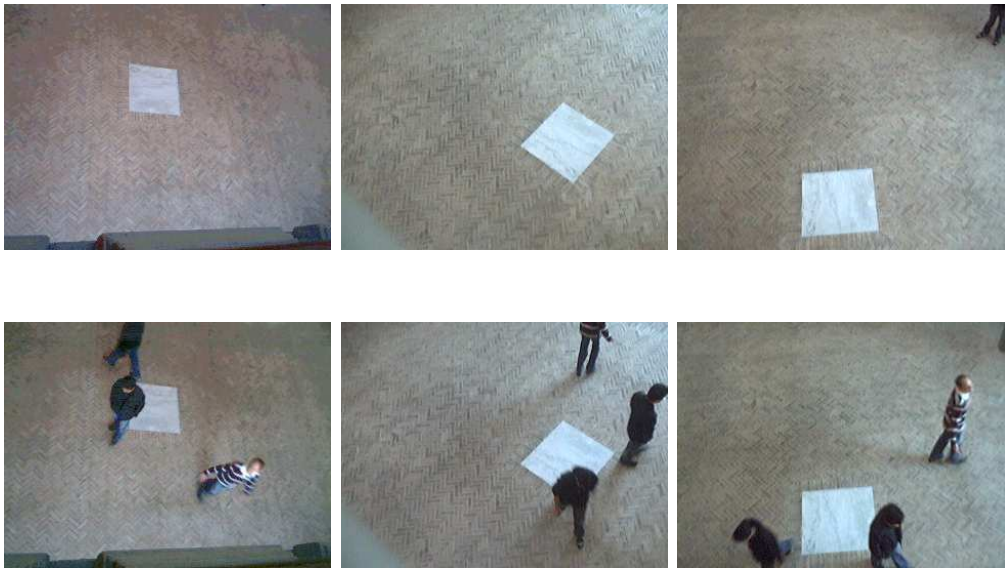


Figure 3.6: Building camera network. (Top Row) Views from the three cameras without objects in the scene. (Bottom Row) Views from the same three cameras observing people walking.

control over the environment, ground truth for the position and orientation of the cameras was obtained using a point light source and using the method cited in the EasyCal Calibration Toolbox [2].

The resulting error in the estimated position, up to scale, can be seen in Figure 3.7 and the estimated orientation error can be seen in Figure 3.8. The center camera's coordinate frame was chosen as the world coordinate frame and the right and left cameras aligned to the center camera's coordinate frame in the global recovery step. It can be seen that the error in the estimation of the localization parameters is small even using the centroids of the objects in the scene.

camera	sequence 1	sequence 2	sequence 3
Right	5.04	6.22	5.81
Left	4.11	7.68	5.91
Center	0	0	0

Figure 3.7: Position Error: The error in the estimated position, up to scale, from the tracks is given in degrees here. The coordinate frame of the center camera is chosen as the world coordinate frame and all other coordinate frames are aligned to this.

camera	sequence 1	sequence 2	sequence 3
Right	1.42	3.67	2.84
Left	2.13	2.56	3.21
Center	0	0	0

Figure 3.8: Orientation Error: The error in the estimated orientation from the tracks is given in degrees here. The coordinate frame of the center camera is chosen as the world coordinate frame and all other coordinate frames are aligned to this.

3.3.4 Outdoor Camera Mote Network with Two Cameras

In this setup two Citrix camera motes were placed in second story windows looking down on a courtyard. The Citrix camera motes were synchronized within a few frames of each other and people walking in the courtyard were observed. This is shown in Figure 3.10.

We positioned two camera motes 8.5 feet apart and pointed them at an open area where people were walking, as shown by the top row of pictures in Figure 3.11. Each camera mote ran background subtraction on its current image and then sent the bounding box coordinates back to the base station for each detected foreground

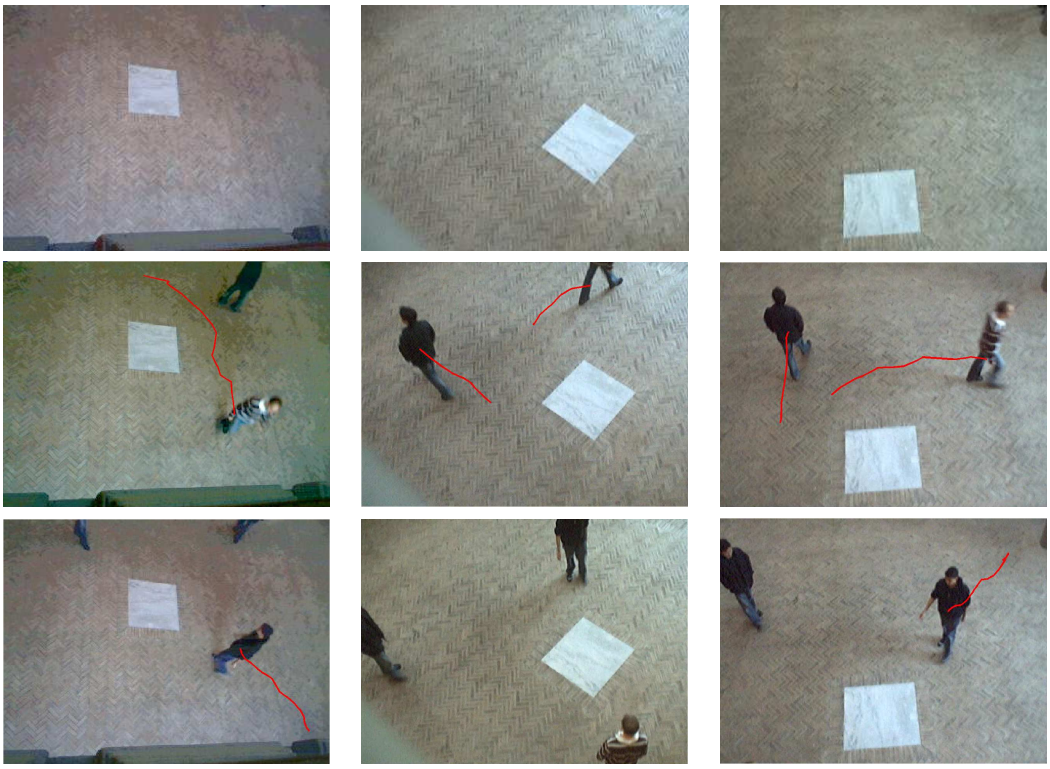


Figure 3.9: Real Camera Network: (Top Row) Images from the cameras with no moving objects. (Middle Row) Images from on data set with tracks of moving objects shown over time. (Bottom Row): Images from on data set with tracks of moving objects shown over time

object. The center of each bounding box was used to build the image tracks over time on the base station computer, as shown in Figure 4.6. It can be seen that multiple tracks are successfully estimated from the image sequence.

Our localization method is particularly suitable for a low-bandwidth camera network because only the coordinates of the foreground objects need to be transmitted, not entire images. In implementing the localization, tracks from the two image se-

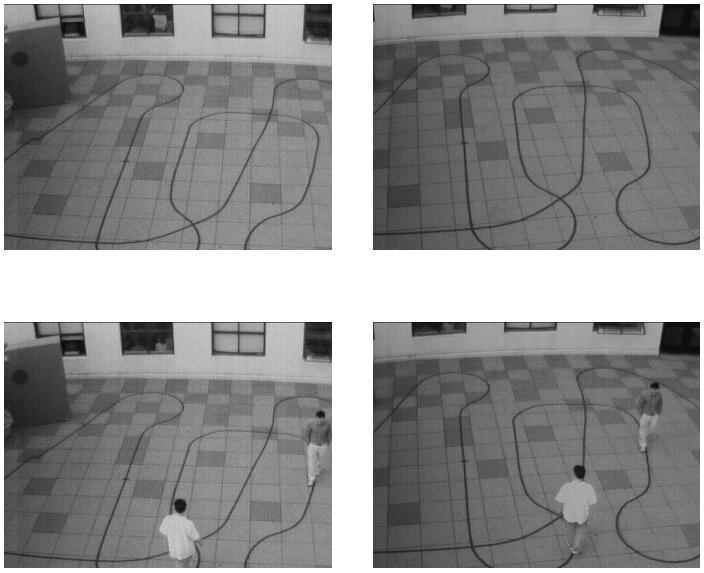


Figure 3.10: Camera Mote Views. (Top Row) View of the courtyard without any moving objects. (Bottom Row) View of the courtyard observing people.

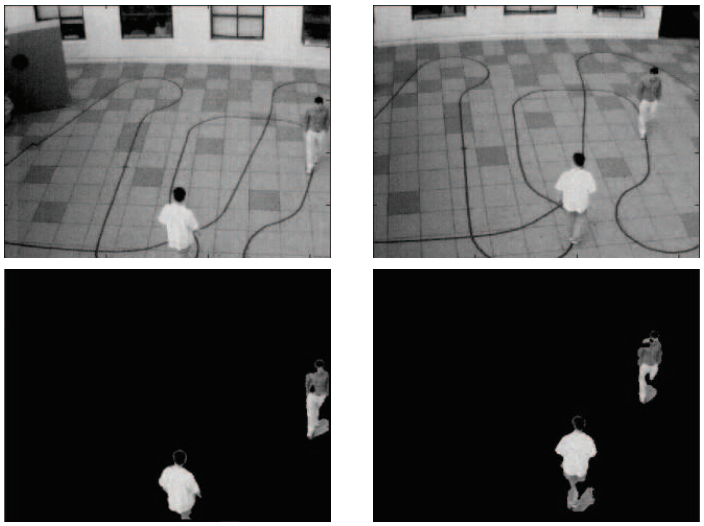


Figure 3.11: (Top) Image frames from the left and right camera motes, respectively, viewing the scene. (Bottom) The detected foreground objects from the scene.

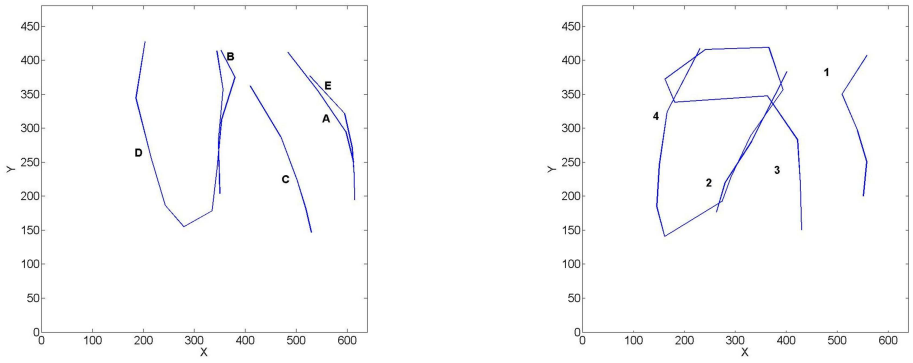


Figure 3.12: The tracks of the moving objects in the image planes of the left and right camera motes, respectively, formed by MCMCDA.

quences are compared, and minimized reprojection error and assumed a common ground plane to determine which tracks best correspond between images. We used 43 frames from the cameras at an image resolution of 640×480. Foreground objects were detected in 22 of the 43 frames and tracks were built off these detected foreground objects. Four tracks were built in the first camera and five tracks were built in the second camera. Using the adapted localization method, we were able to determine the localization of the two cameras relative to one another with an average reprojection error of 4.94 pixels. This was based on the matching of four tracks between the two cameras which minimize the reprojection error.

The accuracy of the camera localization estimate is affected by a few factors. First, the choice of the (low-cost) camera has an effect on the quality of the captured image. Second, the precision of the synchronization between the cameras affects

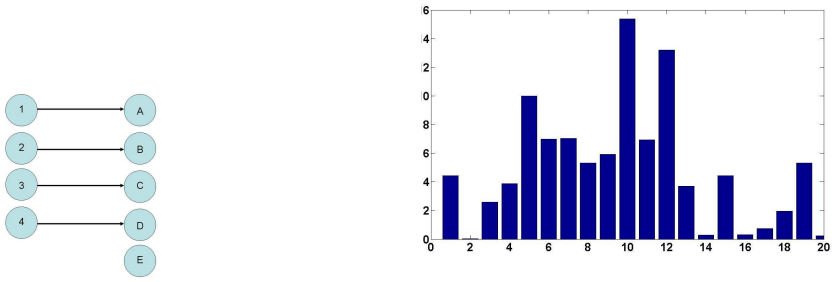


Figure 3.13: (Left) The matching of tracks between the cameras that were used for localization. (Right) The reprojection error measured in pixels for each of the 20 points of the tracks.

the accuracy of the image correspondence as the frames synchronization was off by approximately 2 seconds. Last, we only used a small number of frames to estimate track correspondence. Using a longer image sequence with more data points can reduce the estimation error.

Chapter 4

Full External Calibration Using Data Fusion

In union there is strength.

Aesop

Our method of localizing based on object image tracks provides an automatic way get localization up to a scale factor on the translation vector, i.e., the position of each camera is only recovered up to scale. However, image data alone, no matter what automatic feature is used, is not sufficient to do full external calibration of a camera network in an *unknown* environment because it only recovers the baseline up to scale. In this chapter, we explain a fusion-based method of full external calibration, as first presented in [52], recovering (R, T) and the scale factor λ, for all cameras in the network. By extending our method based on object image tracks and fusing it with radio information, we are able to do recover estimates of the full external parameters

of the camera network. We develop both a linear and a non-linear approach to fuse the image data and the radio interferometry data in order to fully localize the camera network. This Fusion-Based Localization (FBL), or full external calibration, can recover both the orientation of camera's fields of view as well as their complete positions.

4.1 Problem Formulation

Suppose that we have N cameras in the network. We assume that all the cameras are time-synchronized and each camera is equipped with a radio for wireless communication. Each camera is assumed to have the capability of detecting features, F, in the scene as well as having some overlap in what it sees with another camera in the network for correspondence. While feature correspondence in image data can be used to determine the orientation of a camera relative to another camera that overlaps, the position can only be determined up to scale due to the geometry constraints on the epipoles. Due to this, multiple set of possible positions of cameras will results from the same set of images.

Using image data alone, without previous knowledge on the scene or objects moving through the scene, we can localize the cameras except for scale factors, λ_i on the vectors, T_i, that relate to the position of each camera C_i. With three overlapping cameras which see the same moving objects at the same time and with sufficient motion, we can recover the scale factor. However, this is not typical in many networks

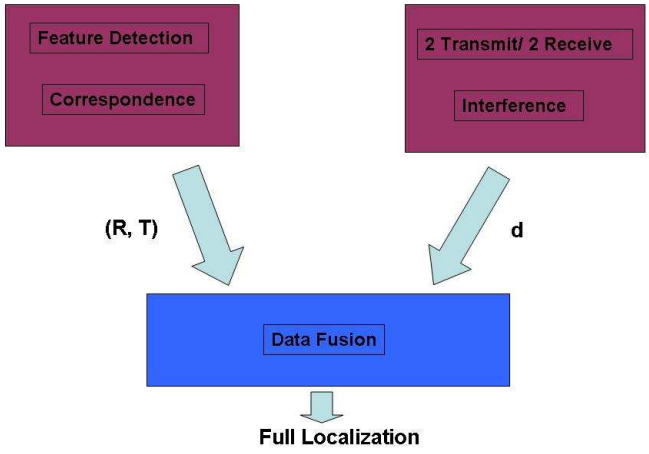


Figure 4.1: The structure of the data fusion localization method.

and puts further constraints on how the cameras in a network can be places. To recover these scale factors, we instead look to incorporating another type data from the low-bandwidth sensor network community. Since radios are often found in incorporated with other sensors in a network and have become the common means for localizing low-bandwidth networks, it is reasonable for us to look to using this type of information to recover the scale factors involved when localizing camera networks.

The omnidirectional radios on a wireless node can transmit and receive data but they do not provide directional information. By communicating with one another, the radio nodes can use signal interference to determine linear combination of distances between groups of nodes. Note that the orientations of cameras can not be solved

using radio data alone.

Our FBL method solves for the complete localization of a camera network, finding both the orientation of the fields of view of each camera as well as the positions of cameras including the scale factor by using q-ranges, $\{d_{ijkl}\}$, computed from the radio interferometry; and the relative orientations, $\{R_{ij}\}$, and positions up to scale, $\{T_{ij}\}$, from the image object image track localization method from Chapter 3. The structure of the FBL method is shown in Figure 4.1. We develop a non-linear fusion method can to find the resulting scale factor so the position of the cameras are fully known. Given additional parameters, I show that a computationally efficient linear fusion method can be used.

4.2 Overview Radio Inferometry

The Radio Interferometric Positioning System (RIPS) was proposed in [48] for node localization using the phase measurements of the radio signals with low cost hardware. The basic idea behind RIPS is to utilize two transmitter nodes to create an interference signal. The two nodes transmit sine waves at slightly different frequencies at the same time, creating a composite interference signal with a low frequency envelope. This interference frequency can be measured by cheap and simple hardware available on a wireless sensor node.

Figure 4.2 shows an example of the interference signal and its low frequency beats at nodes C and D. The model of the radio interference was developed in [48]. The

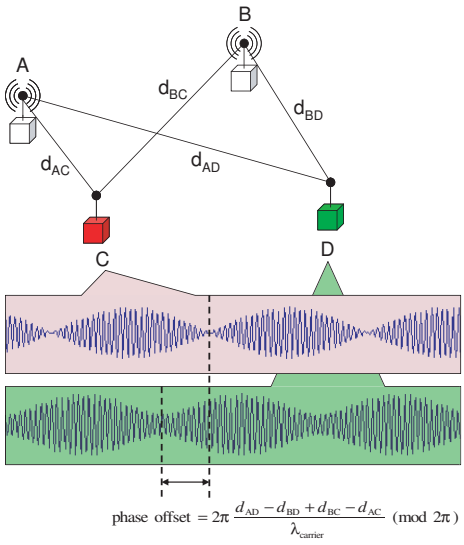


Figure 4.2: Two transmitters A and B transmit at the same time at two close frequencies. The interfere signal is observed by receivers C and D. Figure from [48].

phase offset of the interference signals received at two different receivers can be expressed in terms of a quantity called the *q-range*, which is a linear combination of distances between the two transmitters and two receivers defined as

$$q_{ABCD} = d_{AD} - d_{BD} + d_{BC} - d_{AC}$$

An important theorem on *q-range* presented in [48] states that the relative phase offset of received interference signals at nodes C and D are related to *q-range* as follows

$$\varphi_{CD} = 2\pi \frac{q_{ABCD}}{c/f} \pmod{2\pi}, \tag{4.1}$$

and $f = f_A + f_B$, where f_A, f_B are carrier frequencies of transmitters A and B, and

c is speed of light.

Another important result presented in [41] states that in a network of n wireless nodes, there exist a maximum of $n(n-3)/2$ linearly independent *q-ranges*. It was shown that by taking independent *q-range* measurements for different combinations of four nodes, it is possible to reconstruct the relative location of the nodes. An algorithm and implementation for localization were presented in [48].

The main benefit of RIPS lies in the fact that it does not require any additional hardware because common radio transceivers can be utilized for phase measurements.

By using the *q-ranges* we now have distances that can be used in conjunction with the R_{ij}'s and T_{ij}'s we found from the image data. We now need to combine these two sets of information to determine the position of the cameras.

4.3 Fusion-based Localization

While neither q-ranges alone nor image data alone have enough information to completely localize the cameras, fusing the two sets of information gives us complete localization. Here we present both a linear method and a nonlinear method for solving this data fusion problem. The linear method has a unique solution, but requires certain conditions on the topology of the camera network. If these conditions are not met, the nonlinear method can be used for complete localization.

4.3.1 Linear Method

For a network of N cameras, there exist $N(N-3)/2$ independent q-range equations [41]. We also know, given the network, that there are $N(N-1)/2$ possible pairings of cameras, thus $N(N-1)/2$ pairwise scales on position, λ_{ij}, exist. In addition to the $N(N-3)/2$ independent q-range equations, N more independent equations are necessary in order to solve for all the λ_{ij}. We look to the camera position vectors, T_{ij}, where $i \neq j \in 1, ..., k$ and where $k \leq N$ to provide us with these additional equations.

Using the available T_{ij}, we want to find equations such that we can use the vector notation and the unknown scales to write one T_{ij}, with it's unknown scale λ_{ij}, as a sum of T_{kl} and λ_{kl} and T_{mn} and λ_{mn} such that $ij \neq kl \neq mn$. For example, Figure 4.3, the unit translation vector T_{13} can be written as:

$$\lambda_{13}T_{13} = \lambda_{12}T_{12} + \lambda_{23}T_{23} \tag{4.2}$$

If we take this equation and write it in terms of the unknown scales, λ_{ij} we get:

$$[T_{12} \quad T_{12} \quad -T_{13}] \begin{bmatrix} \lambda_{12} \\ \lambda_{23} \\ \lambda_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.3}$$

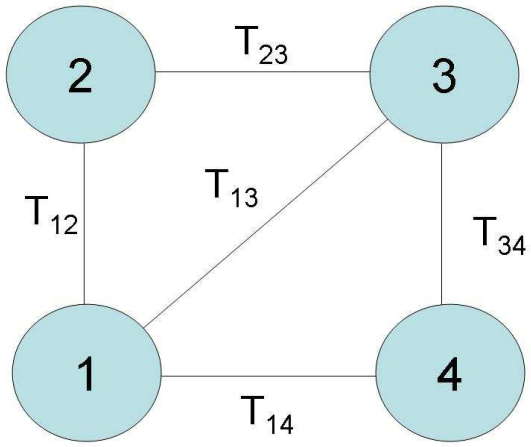


Figure 4.3: The overlap in fields of view between the cameras nodes based on the object image tracks

As can be seen, the matrix $[T_{12} \;\; T_{23} \;\; -T_{13}]$ is not full rank and $[\lambda_{12} \;\; \lambda_{23} \;\; \lambda_{13}]^T$ lies in the null space of the matrix. Thus, we can only get at most, two independent equations from using Equation 4.3. Additionally, it can be seen that a clique of three cameras all being able to view one another, is needed to write this type of equation, in order to write one T_{ij} in terms of other T_{ij}. Therefore, in addition to the set of independent q-range equations, we need N equations from the camera network which means $N/2$ cliques of three cameras need to exist in the camera network.

If enough cliques are found in the camera network, then we can write the unknown scales, λ_{ij} in terms of the q-ranges and translations T_{ij} as a linear system:

$$Ax = b \tag{4.4}$$

where

$$b = \begin{bmatrix} d_{1234} & d_{1324} & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0 & -1 & 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ T_{12} & -T_{13} & \mathbf{0} & T_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T_{13} & -T_{14} & \mathbf{0} & \mathbf{0} & T_{34} \end{bmatrix}$$

$$x = \begin{bmatrix} \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{23} & \lambda_{24} & \lambda_{34} \end{bmatrix}^T$$

Thus, given enough cliques of three cameras in a network which have overlapping fields of view such that all T_{ij} can be found for that clique, a linear method exists to solve for the scales λ_{ij} as shown in Equation 4.4.

4.3.2 Nonlinear

A unique solution can be found from the linear method if certain conditions exist in the camera network. However, when these conditions are not present, we must

rely on a nonlinear method. We have developed a nonlinear method to solve for the external calibration parameters in a general camera network setup.

From the localization algorithm described in Chapter 3 we have a set of pairs of camera nodes with overlapping field-of-views (FoV). Lets denote the set as,

$$\mathcal{O} = \{(C_i, C_j) : C_i \text{ has overlapping FoV with } C_j, i \neq j\}$$

For a camera pair $o_k = (C_i, C_j) \in \mathcal{O}$, denote scaling factor between camera nodes C_i and C_j as λ_k. We also know the relative rotation matrix and unit translation vector for the pair. Let the rotation matrix denoted as R_{ij}^i and the unit translation vector as, $\mathbf{T}_{ij}^i$. Assuming the camera network is connected, we can compute the rotation matrices for each of the camera nodes in a common frame of reference. Without loss of generality, lets consider reference frame of node 1 as the global reference frame. With successive multiplications of the relative rotation matrices we can compute absolute rotation matrices R_{i1} for $i > 1$ as:

$$R_{i1} = R_{i_1 1} \times R_{i_2 i_1} \times \cdots \times R_{i i_n}$$

where $\{1, i_1, i_2, \cdots, i_n, i\}$ is a path from node 1 to node i. For $i = 1$ the rotation matrix is an identity matrix of size 3.

Using the unit translation vectors, absolute rotation matrices and the scaling

factor for each pair, we can compute vector for pair o_k in the frame of 1 as:

$$\mathbf{x}_{ij}^1 = R_{i1} \times \mathbf{T}_{ij}^i \times \lambda_k$$

The camera node locations in the global reference frame can be computed as:

$$\mathbf{x}_i^1 = \mathbf{x}_1 + \mathbf{x}_{1i_1}^1 + \mathbf{x}_{i_1 i_2}^1 + \cdots + \mathbf{x}_{i_n i}^1$$

where $\{1, i_1, i_2, \cdots, i_n, i\}$ is a path from node 1 to node i, and $\mathbf{x}_1 = [0, 0, 0]^T$.

Both vectors $\mathbf{x}_{ij}^1$ and $\mathbf{x}_i^1$ are functions of the scaling factors λ, i.e. $\mathbf{x}_{ij}^1 = R_{i1} \times \mathbf{T}_{ij}^i \times \lambda_k \triangleq \mathbf{F}_{ij}(\lambda_k)$, and $\mathbf{x}_i^1 \triangleq \mathbf{G}_i(\lambda)$. Hence, the distance between all other camera pairs that are not members of the set $\mathcal{O}$ can also be represented as a function of scaling factors as:

$$d_{ij} = ||\mathbf{G}_i(\lambda) - \mathbf{G}_j(\lambda)|| \triangleq H_{ij}(\lambda)$$

The q-ranges defined in section 4.2, which are linear combinations of distances can

also be expressed as functions of scaling factors as:

$$\begin{aligned} q_{\text{ABCD}} &= d_{\text{AD}} - d_{\text{AC}} + d_{\text{BC}} - d_{\text{BD}} \\ &= H_{\text{AD}}(\lambda) - H_{\text{AC}}(\lambda) + H_{\text{BC}}(\lambda) - H_{\text{BD}}(\lambda) \\ &\triangleq Q_{\text{ABCD}}(\lambda) \end{aligned}$$

The problem can be expressed as a non-linear least squares problem as:

$$\text{minimize } \mathcal{L}(\lambda) = \sum_{\text{ABCD}\in\mathcal{T}} (Q_{\text{ABCD}}(\lambda) - \tilde{q}_{\text{ABCD}})^2$$

where $\mathcal{T}$ is the set of q-range tuples, $\tilde{q}_{\text{ABCD}}$ is the measured q-range for tuple ABCD.

4.4 Experiments using Outdoor Camera Network with Six Cameras

In this section, we apply our fusion method on real data. The algorithm is tested on an outdoor deployment of a network of cameras. Using both the camera images and the radio data we are able to estimate the position and orientation of the cameras. The network consists of Linux PCs equipped with Logitech Quick Cam Pro 4000 cameras, and XSM wireless sensor motes. Six camera nodes and 7 XSM motes are used. The ground truth location of the cameras+XSM motes is show in fig 4.4. The cameras have a resolution of 240 x 320 pixels and acquired images at 8 frames per

second(fps). 12 minutes of data is taken from the cameras for localization. Multiple types of objects moved through the scene during this recording from planes to people. An example of the different objects is shown in Figure 4.5

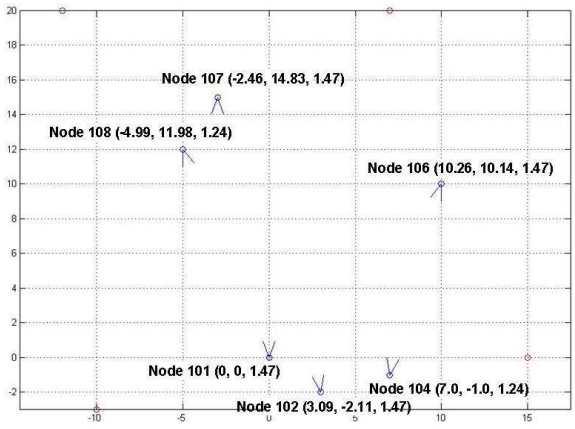


Figure 4.4: An overhead view of the layout of the cameras

We use the existing TinyOS implementation of RIPS developed at Vanderbilt. The TinyOS implementation running on 7 XSM motes and a Java application running on base station provide us with the required q-range measurements. For more detail of RIPS and its implementation we refer reader to [41].

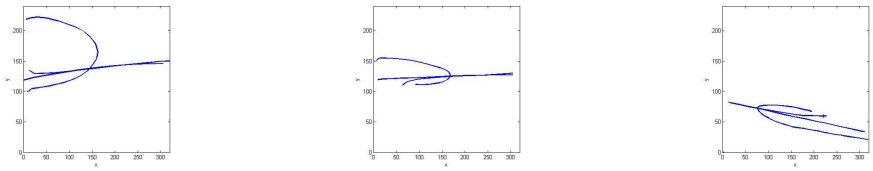
The steps of the automatic method applied are as follows. Adaptive background subtraction is applied to each image to obtain the foreground objects. Bounding boxes are created around the foreground object and if a bounding boxe is located at the edge of an image, the foreground object is not considered for further processing to build object image track. The reason this check is implemented in the algorithm

Figure 4.5: Some of the moving foreground objects in the scene as observed from camera 101

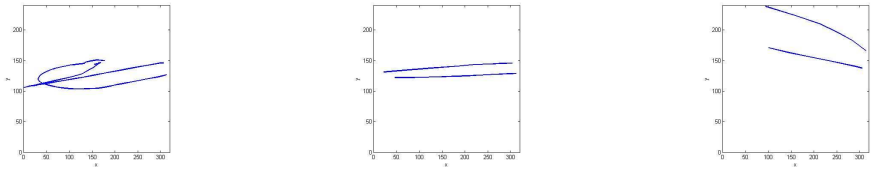
is because a bounding box at the edge of the image could indicate that part of the foreground objects it getting occluded and thus the centroid would not be unstable.

Using the remaining centroids from the bounding boxes, object image tracks are built using MCMCDA. In our method, the number tracks needed is defined to be at least 4 seconds long to further constrain the paths used and so each track had to have at least 32 points due to the frame rate. This parameter in the algorithm can be adjusted as desired.

The tracks from the centroids are then used for feature correspondence between the cameras using the epipolar constraint and R and T, up to scale, solved using SVD. Figure 4.7 shows what cameras are visually connected given the correspondence



Camera 101 Camera 102 Camera 107



Camera 108 Camera 104 Camera 106

Figure 4.6: (Top) Object image tracks for frames 1 through 500 for cameras which had correspondence in their respective tracks (Bottom) Object image tracks for frames 1 through 500 for cameras that did not have correspondence with the top row cameras, but which had correspondence with each other.

between object image tracks. As can be seen, cameras 104 and 106 are disjoint from the rest of the network. While their fields of view overlap with each other, they are not connected to the rest of the network. We verify this is correct based on the ground truth of the setup. The average reprojection error for all camera pairs is < 3 pixels.

Using the T_{ij}'s for all overlapping camera pairs, this is fed into both an automatic nonlinear and linear methods along with the q-ranges obtained from the XSM motes. The resulting scale factors are shown in Table 4.1.

It should be noted that linear method solves for all scaling factors even if the camera pair does not have overlapping field-of-view.

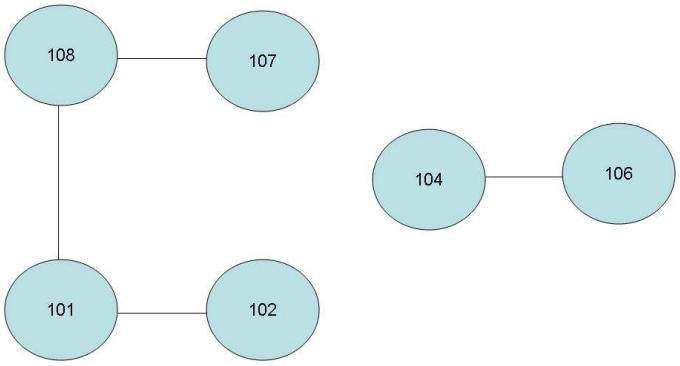


Figure 4.7: The overlap in fields of view between the cameras nodes based on the object image tracks

Camera Pair	Ground Truth	Nonlinear Estimate	Linear Estimate
101 to 102	3.7417	3.2376	4.2593
101 to 107	15.0326	15.5074	13.5940
101 to 108	12.9797	13.2370	13.1016
102 to 107	17.8260	15.4361	14.9415
102 to 108	16.2440	–	15.4440
107 to 108	3.8179	3.6743	1.5795

Table 4.1: The scale factors for the distances between cameras.

Chapter 5

Localization Discussion

The eye sees what it brings the power to see.

Thomas Carlyle (1795-1881)

Our work is one step toward using dynamic information to do data correlation in camera networks. By localizing the cameras in a network, this can aid as stepping stone for higher level algorithms to take advantage of to better understand the visual data in the network and operate the network more efficiently. In this final chapter, we discuss how if there is more information known ahead of time about the network setup, how this can be taken into account in our localization method. Additionally, we discuss further steps towards data correlation in these camera networks.

5.1 Incorporating Known Parameters

The continuing focus of this localization work is that it is important and how to provide a general method that is not restricted to specific camera network setups or known scene information. Yet, our method can easily be adapted to take into account more known parameters if this information is present. Here we give an overview of some of the varying parameters that can be known ahead of time about the camera network or the environment and how these fit in to our general method for localization.

Single Object: If it is known ahead of time that only a single object will be moving through the environment, this simplifies the inter-camera track matching and correspondence of the localization method. When the object is a known objects with known features, this additionally simplifies the intra-camera track formation portion of the method.

An instance where a single object may occur is when a known object is placed into the scene, for example a robot, which navigates and is tracked given known features. While there may be other moving objects in the scene during this time, only the known object will be detected, based on the known features of this object, and considered for track formation. This simplifies the intra-camera track formation step as more constraints are put on detection moving objects to be considered for track formation. Only the known object is detected, based on it's features, for track formation based and tracks are formed off this single object.

Another example of when a single object might occur is when it is known ahead

of time that there is only one object moving in through the scene due to specific scene constraints, such as a narrow space or patterns of motion. While this object was not placed in to the scene and known a priori, it is known that there is only one object and this again simplifies the localization algorithm as there is only one object considered as a moving object for the method.

If there is only one object considered to be a moving object for the localization method, then the intra-camera track formation only builds tracks of this single object. The inter-camera track matching and correlation is then greatly simplified. The process $\Psi : (\Theta_i, \Theta_j)$, does not have to be run as the track matching is a given as long as the tracks from different cameras have an overlap in time. Additionally, when tracks overlap in time, we know they are a correct match, as they are corresponding to the same moving object, thus the $\Omega(E_m)$ process does not need to be done. The essential matrix E can just bet determined straight away from the correspondences formed from the overlapping tracks.

Global Ground Plane: When the scene structure is known ahead of time and it is a given that a global ground plane exists between cameras, then this can be taken into account in the inter-camera matching and correlation step. Instead of using $\Omega(E_m)$ and looking at the space of essential matrices, since there is a global ground plane we can instead look for a homography between the cameras instead of the essential matrix [46]. All processes in the inter-camera matching and correlation step that deal with the essential matrix will instead deal with the homography instead.

We demonstrated this in our experimental results section with in 3.3.2.

Limited Static Features: When a set of static features and their correspondences between cameras is known, yet these are not enough to solve for the localization of the cameras, these can be used in the inter-camera matching and correspondence portion of the method. Since the correspondences of these static features is known, they must be upheld by the essential matrix. In the inter-camera matching and correlation step, once an estimate, E_m, this estimate can be used to calculate the reprojection error for the static correspondences. Since these static correspondences are known, an essential matrix E_m which minimized the reprojection error on these correspondences as well as the tracks is desirable.

5.2 Toward Data Correlation

Localizing the cameras is only one aspect of data correlation with dynamic scenes. Localization gives the position and orientation of the fields of views of the cameras in the network and a model of how the sensing fields relate which can be used for further issues in data correlation. For example, localization information helps when determining motion patterns between cameras from the dynamic scenes. This is an aspect of data correlation that is used in applications such as tracking and anomoly detection. By determining how moving objects transition between cameras, a model or distribution can be built for traffic patterns in the network.

If two cameras have an overlapping in their fields of view where an object transition

through, then it is easier to determine a model for object motion between these cameras. At this point of overlap, where the object is seen by both cameras, the localization gives and a geometric relation for this object on how it should look in each camera. The object may have different appearances in each camera's field of view, yet due to the geometric constraint at the overlap, it can still be determined that this is the same object seen by both cameras. For the areas where the cameras do not overlap, the similarity in appearance of the object within each camera's field can be used to track the object and when the object reaches the overlapping region, the non-overlapping information from each camera can be related to each other through this overlap.

With our method, we have taken a good first step toward improving data correlation in camera networks. We have demonstrated a novel technique for doing localization of cameras in a network using dynamic scene information. We have demonstrated how this technique work on both simulated and real data and shown that using moving objects can give a good estimates of the localization parameters, (R, T), in situations where static features cannot be used. Additionally, we have discussed the limits of image data and how it can be bolstered by radio data in the network to recover all extrinsic parameters. We hope that our work will inspire others to go beyond single frame information and explore how dynamic scene information can be used.

Chapter 6

Introduction to Camera Network Attack Detection

The ultimate security is your understanding of reality.

H. Stanley Judd (1936-)

In recent years, the growth of camera networks has progressed at a rapid pace, particularly after the events of September 11, 2001. Camera networks have drawn increasing attention and their use has become more widespread, particularly in public, uncontrolled spaces. The images from the cameras provide a rich source of information about the environment and the distributed nature of the sensor network introduces flexibility in system scaling as well as different perspectives of the same event or subjects of interest. Yet, the expansion in use of these networks raise security concerns as more often these cameras are in space where they are open to attacks. While there are measures to protect image data in transmission and storage, detecting attacks on

the network to determine if the data being transmitted is trustworthy is still an open area. In this chapter, we give brief overview of current security measures for camera networks and the current challenge of detecting attacks on the cameras themselves. We discuss how taking advantage of the image data can be used for detecting attacks on the cameras in the network.

6.1 Camera Network Security and the Importance of Image Data

In any network, it is important to consider security concerns in order to prevent substantial performance degradation or privacy intrusion. Particularly in camera networks, with the level of detail that image data provides, security breaches can have a large impact. For example, in Maryland in 2003, three women were accidently arrested and charged with murder because of an unsynchronized camera at an ATM. Their pictures were flashed in front of a national audience and they spent three weeks in a Maryland jail before it was discovered that the camera was set to the wrong time [84]. In 2006, a German museum's security camera on the roof of the building was moved without authorization to spy on the private Berlin flat of German Chancellor Angela Merkel [27]. In 2006, an Austrian group called Quintessenz, hacked into police video feeds in Vienna Austria. The group used an off-the-shelf 1-GHz satellite receiver to intercept the video signal transmitted by a surveillance camera overlooking a busy

square in the Vienna. This enabled them to view everything recorded by the camera, which was only authorized to be seen by the police[32].

A security breech can occur at any point in the camera network. Methods exist for detecting and protecting data from security breeches at the base station and while in transmission. At the base station where image data is stored, access control methods can be used to allow only authorized entities to view the image data. For example, a base-station can handshake with a user and make the user prove that it has a correct cryptographic key or password for viewing the image data. Stanton et al. evaluates a number of different multimedia storage solutions [82].

During transmission, image data can be protected and security breeches detected via methods such as encryption, watermarking, and message authentication Codes (MAC). Data encryption has shown to be effective at preventing security breeches and custom-tailored algorithms have been designed which exploit properties of image/video compression protocols, such as MPEG, and are described by Shi et. al [80] and Li et. al [44]. Semi-fragile watermarking [67, 70] and image hashing functions [95, 73] can also be used so the base station can authenticate the image data and ensure it has not been corrupted. Performing authentication to verify that the data is coming from a specific camera node, rather than an attacker inserting data can be accomplished using MACs on the communicated data.

While the transmission and base station portions of the network have security methods available to them, limited work has been done on methods for detecting

security breeches at the camera node portion of the network. In particulary, doing intrusion detection on the camera node end of the network is an open area. Intrusion detection is defined as detecting a deliberate unauthorized attempt or attack by an unauthorized entity to access information, manipulate information, or render a system unrealiable or unusuable [8]. In doing this intrusion detection on the camera in the network to determine if an attack has happened on the data before it was transmitted, specific image-based security methods must be considered.

Attacks on the cameras affect the image data the camera is going to send through the network, for example, changing where the field of view of the camera is pointed is one type of attack. As these attacks may not affect any other function of the network, the image data must be relied upon to provide the necessary cues to determine if an attack has occurred. While this could be done manually, by constantly monitoring the images getting transmitted, often times the data from the network is not monitored in real-time and there is too much information coming in for a human to detect when an attack has occurred. Additionally, if the images from the cameras are not actively monitored, just transmitted and then stored so they can be used at a later date, manual methods will not work. It is still necessary to know, when this image data is recalled, if it is faulty or not. Thus, it is important then to have an automatic detection method, based on image data, for determining whether a camera has been compromised and if its image data is faulty.

6.2 Related Work for Attack Detection

In doing intrusion detection on cameras in a network, we propose a method that uses image features to build a reputation of the cameras to determine if they are sending faulty data or not. There is a large body of work on reputation systems as they have proven useful as a self-policing mechanism to address the threat of compromised entities. The idea has been used in many fields, including, economics, sociology, and computer science. For the sake of space, we discuss some of the key works where the reputation systems have been applied, particularly in sensor networks.

Centralized reputations systems were popularized by the internet [24, 76, 68]. An example of this in practice is Ebay's rating system [76]. Decentralized methods for use in self-organized networks were discussed for use in [59]. Applications of this to sensor networks include the *CORE* reputation system [54] and the CONFIDANT protocol [13]. These methods use a watchdog module at each node to monitor the forwarding rate of the neighbors of a node. If the node does not forward the message, its reputation decreases, and this information is propagated throughout the network. The watchdog module also uses the second hand information from other nodes to find the overall reputation of the node. Over time the badly behaving nodes are less trusted and will not be used in forming reliable paths for routing purposes.

The most relevant work to ours is presented in [28]. The authors suggest a high level reputation system framework for sensor networks. The main difference between our work and [28] is that the authors only suggest a 'watchdog' mechanism to de-

termine the reputation of each node. They state that there is no unifying way in which reputations can be assigned, i.e. the mechanism to assign reputation has to be context dependent.

While the idea of reputation is very prevalent and proven useful in detecting attacks, the works dealing with detecting attacks on cameras have not explored the idea and are more ad-hoc. Works discussing attacks on cameras term the issue as camera tampering with varying definitions. In [7], tampering is defined as the camera lens being obscured, for example by a foreign object or spraying paint, or if the camera has been defocused is discussed. The detect tampering, an intensity description, in the wavelet domain, of the background of the scene the camera is observing is learned. Camera tampering is deemed to have occurred if the current frame's intensity varies over a given threshold from the learned background intensity values. Additionally, the current frame's intensity must be darker than the background scene as it is assumed that obscuring the camera forces intensity to decrease.

Alternatively, tampering is defined as any sustained event which dramatically alters the image seen by the camera in [78]. Here the authors present a method based on comparing the color histograms of current frames to older frames in different manners. They compare the gradient of the histogram, the L1 distance between histograms, and the distance between the two histograms. If all three aspects vary, it is determined that an attack has occurred.

6.3 Contributions of Our Approach

With our intrusion detection method, we build upon the previous works, using both the idea of reputation and the idea of image-based analysis to determine if an attack has occurred on each camera. However, we combine the two ideas in order to do intrusion detection on the cameras. Using the idea of reputation, we are able to move away from ad-hoc image tampering methods and use a statistically founded approach to intrusion detection. We do not focus on physical attacks to the cameras alone, but expand our method to deal with more invasive attacks where the image data has been altered, but the camera has not been moved or obscured. Additionally, we do not rely on intensity or color data alone, as these aspects are prone to change just due to a noisy environment and not strictly due to an attack. As cameras are often monitoring dynamic scenes, it is important our method is robust to this and color and intensity information does not lend itself to that. We discuss what features from image data can be used and are more robust to the environments camera networks are in. We expand the idea of reputation to deal with multi-modal data and how this can help define attack signatures. We also develop an automatic assignment mechanism for reputation for the cameras and do not rely on watchdog modules or context. Our approach is flexible and can be used in varying setups.

We create an automatic image-based method to detect attacks on cameras in a network and determine whether they have been compromised is presented. By analyzing geometric constraints as well as differing camera network setups, we determine

what features from the image data can be used as indications of attacks. As cameras are a rich source of information, there are multiple types of features that can be pulled from the image data and we explore what features can be used and how to treat them in a dynamic scene such that they are robust to environmental noise. We then demonstrate how these features can be used in a reputation scheme in order to detect attacks on the cameras.

Chapter 7

Attack Detection Using Image-Based Reputation

The way to gain a good reputation is to endeavor to be what you desire to appear.

Socrates (469 B.C. - 399 B.C.)

To do automatic intrusion detection on the cameras to determine when they have been attacked and are giving faulty information, we look at varying types of features and how they can be used in a spatio-temporal comparison for reputation. The reputation of each camera will then let the network know whether the camera is trustworthy, and sending correct information, or whether the camera is untrustworthy and sending faulty information. In this chapter, we describe our method of detecting attacks on cameras using image-based reputation. We present the necessary assumptions on the network in order to use this method and describe the different

features that can be used, how they are correlated, and in what camera setups they can be used in order to determine reputation.

7.1 Problem Formulation

A Camera network consists of some cameras, positioned out in the environment to be observed, and a base station. There is a supporting communication infrastructure such that the camera nodes can communicate to the base station and to each other. The cameras are the nodes of the network which obtain data about the observed environment and then communicate this to the base station as shown in figure 7.1.

7.1.1 Network Setup

The camera network setup is as follows:

- The cameras are static and at fixed locations in the environment
- The camera network's initial setup is done without any tampering by an adversary. Thus, the location, including position and orientation, of cameras, synchronization, and communication with the base station has been initialized correctly.
- Once this setup is done, we assume only a trustworthy base station, but make no other assumptions about the cameras or the communication channel.

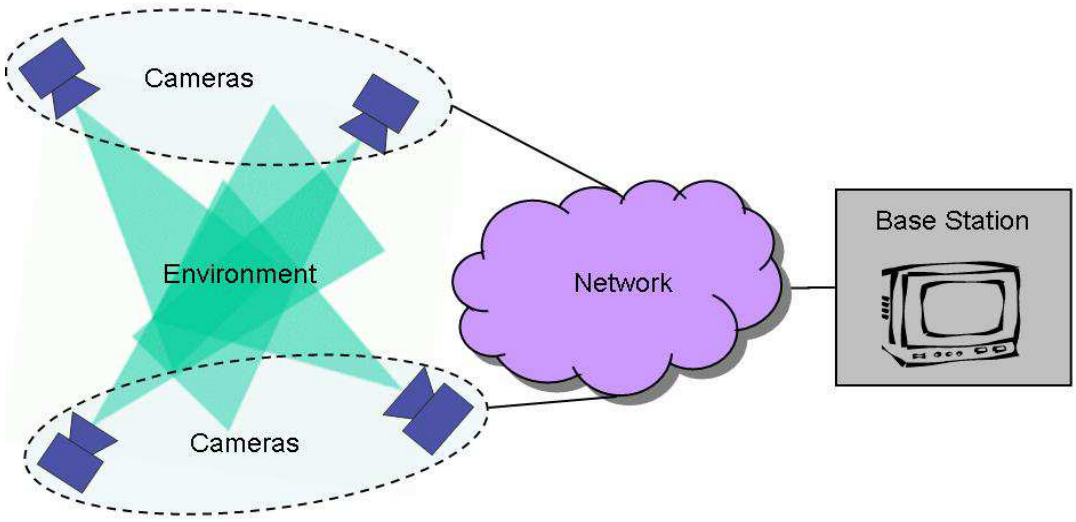


Figure 7.1: Camera Network Setup.

- Cameras capture 10 or more frames per second

We make no assumption on what types of cameras are used, beyond the fact that they are static, nor what communication method, whether it be IP based wireless transmission or through coaxial cable. With the above system description, our input and output are as follows:

- **Input:** Synchronized image data from the cameras in the network
- **Output:** Detect if an attack has happened at the camera node end of the network and which, if any, nodes have been tampered with and are sending

faulty information.

The network has n cameras with a possibility of one or more cameras being attacked and sending faulty data. There are three different types of camera configurations that can exist in the network as shown in Figure 7.1.1. We focus on these three different configurations as different combinations of them cover all possible camera network setups. These arrangements are as follows:

- **Single Camera with No Overlap** In this arrangement a camera has no overlap in its fields of view with any other cameras in the network.
- **Multiple Overlapping Cameras** In this arrangement, there are at least 2 or more cameras that have an overlap in their field of views. There can be multiple groups of camera with overlaps in at least pairwise fields of view.
- **Multiple Nonoverlapping Cameras** In this setup, at least two cameras have a correlation in the spatio-temporal domain in the data they see even though they do not have overlapping fields of view.

7.1.2 Types of Attacks

For intrusion detection on the camera nodes, the attacks we are trying to detect are active attacks since the adversary effects the functioning of the network. An active attack is one that interferes with the operation of the network in some way

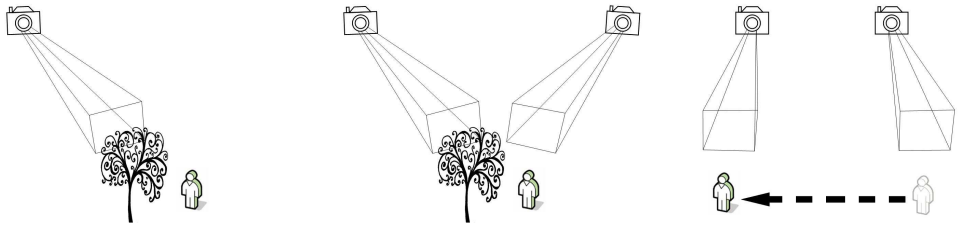


Figure 7.2: Three different camera setups: (a) a single non-overlapping camera (b) two cameras with overlapping fields of view and (c) two cameras with non-overlapping fields of view, but correlation on motion between them.

as opposed to a passive attack in which the adversary just listens to or eavesdrops on network traffic. Here we look at the following active attacks that happen on the camera nodes.

Non-invasive

In this type of attack the internal workings of the camera are not tampered with nor the algorithms running on it. However, external operations, such as the location of the camera, can be altered. Non-invasive attacks involve the least skill from the adversary, but can still have an significant impact on the network operation. These types of attacks include:

- **Moving the Camera:** This involves moving a camera to a different location than where it was originally designated to be when the camera network was setup . If a specific area is suppose to be monitored by a camera node, moving this node will change the coverage area affecting the purpose of the system. For

example, if the camera network is use for surveillance, moving a camera may leave an area unmonitored and create a vulnerability for the attacker to escape detection by the system. If image data from cameras with overlapping fields of view is needed in order to do some task, for example 3D reconstruction, moving a camera will result in the inaccurate results.

- **Blocking the Camera:** This involves putting an object in front of a camera to occlude part or all of the field of view or just covering up the camera lens completely. For example, people have blinded cameras with balloons, lasers and infrared devices, and put covers over the camera lenses [32, 37]. These kinds of attacks, while not necessarily preventable, can at least be detected by computer vision techniques so appropriate steps can be taken.

- **Defocusing the Camera:** This involves changing the focus level of the camera to make the image blurry so that no details can be determined.

The effects of non-invasive attacks vary depending on the network design. It can make an area that the network is originally supposed to be monitoring vulnerable because there is no longer image data coming from that designated area. It can affect distributed algorithms that depend on the location of the cameras. For example, if there is camera handoff so that an item being tracked can be followed throughout the coverage space of the camera network, the handoff will not happen correctly or no information will come from the attacked camera. When overlapping fields of view

from multiple cameras are needed to perform certain tasks like 3D reconstruction and depth calculations, then if one of those cameras is attacked, these calculations cannot be performed. Thus, it is important to find ways to detect when these types of events are occurring and trigger and appropriate response.

Invasive

Invasive attacks involve changing the actual data the camera node is sending by changing the internal working of the camera node This type of attack requires a more skillful adversary than a non-invasive attack as it may entail reverse engineering followed by probing techniques and access to the chip level components of the camera. As a result, the attacker has an access to the information stored on the chip, and can cause substantial damage to the system. Invasive attacks include:

- **Time-stamping Change:**Changing the time-stamping on images from a camera by altering its internal clock is just one example. If the cameras are synchronized, and the attacker is able to change the time stamping of one camera, then the frames corresponding to the same global time instance will be different on two neighboring cameras. This can effect tasks of the network, for example tracking of a moving object through a surveillance area. If one camera node sees the object and sends a signal to a camera node with a compromised clock to pick up the object at a certain time, this triggering will not work and the object may not get tracked appropriately. A method for re- synchronizing the cameras must be developed in order to alleviate this type of attack. This can

be done using a signal sent out from the base station or if that is not possible, synchronize across nearby cameras using a common feature shared in time.

- **Injecting Faulty Data:** This type of attack involves using the camera to send faulty data that the attacker injects into the system and which the camera node would never have obtained. This includes replay attacks. A replay attack involves an adversary delaying a feed from a camera node or putting it in a play back loop of some sequence of images that the camera previously captured. The images being fed back to the base station will not actually correspond to what is going on in the environment at the current time. However, the time-stamping is correct and the images were taken by the actual camera at that location, just at a different time. Thus, it will be hard to detect this attack as they images may look correct and the time-stamping and authentication codes will be correct.

Both non-invasive and invasive attacks can be hard to detect even if images from the camera network are being actively monitored at the base station by authorized entities. It is difficult for humans to take in and process in real time the amount of visual information that can be coming in from multiple cameras . Cues that a camera node is being tampered with can be missed as the cue may only take up a few frames or the differences may not be immediately apparent to the human eye. Thus, an automatic method is needed that can more easily and quickly detect these attacks on the camera nodes, yet still be robust to the dynamic environment which the cameras are observing and noisy data. This is what our method based on reputation systems

provides.

Our reputation method will provide notification of when a camera node is being attacked while still being robust to the environmental conditions and noisy data. If a camera network is being actively monitored at the base-station, then an appropriate entity can respond to the notification of tampering. However, if the network is not being actively monitored, then the notification can be used to mark data coming from that camera as faulty so that if the data recorded by the network is needed at a later date, faulty data will not be used.

7.2 Beta-Reputation Expanded

For a camera network, we want to determine which camera nodes are are untrustworthy. In order to do this, we look at feature observations from the image data of the cameras and assign a reputation based on how consisted these features are. For a given camera, we assign reputation based on the consistency of features with 1) previous features from the same camera and/or 2) features from other cameras which have a relation to the given camera.

To assign a reputation to a camera, we use the Beta distribution based reputation [35]. When dealing with binary event outcomes the posterior probability of binary events can be represented as a Beta distribution. Since observations of an image feature can be thought of as a binary event, the Beta distribution is well suited for our purposes. Define a binary even with an outcome of either y or $\bar{y}$. The Beta

probability density function then takes the integer number of past observations of y and $\bar{y}$, which we define as s and $\bar{s}$ respectively, to predict relative frequency with which y will happen in the future. The Beta distribution is the conjugate prior of the binomial distribution and the expected value of the Beta distribution then gives us an the expected relative frequency with which y will happen in the future as show below:

$$E(p) = \alpha/(\alpha + \beta) \tag{7.1}$$

where

$$\begin{aligned} \alpha &= s + 1 > 0 \\ \beta &= \bar{s} + 1 > 0 \\ 0 &\leq p \leq 1 \end{aligned} \tag{7.2}$$

and p is a probability variable representing the probability of an event, in this case of y, occuring.

Mapping this now to the reputation of cameras, we can treat the appearance of a feature at a given time instance as a binary event. At any time instance, a feature is either seen or not seen and thus, we can treat the degree of trustworthiness/untrustworthiness as a based on the number of binary outcomes of each feature observation. The pair of variable $(s, \bar{s})$ is a continuous pair of values that now re-

lates the degree of trustworthiness, s, and untrustworthiness, $\bar{s}$, based on the binary feature observations. The posterior probability of this type of continuous variables is most accurately represented by the Beta distribution and has been shown mathematically in [35]. In Bayesian statistics, the Beta distribution can then be seen as the posterior distribution of parameter p of a binomial distribution after observing $\alpha - 1$ independent events with probability p and $\beta - 1$ with probability $1 - p$. This type of reputation scheme is beneficial because it is simple, flexible, and relies on statistical theory instead of ad-hoc rules as previous methods in other areas have done before.

In order to determine the overall reputation of a camera using the Beta distribution estimate, there are three main steps:

- **Instantaneous Rating:** At each time instance, positive and negative ratings are assigned to a camera node based on the outcome of binary feature observations.

- **Aggregation:** The instantaneous positive and negative ratings are aggregated over time for robustness.

- **Reputation Assignment:** The aggregated positive and negative ratings are combined using the expected value of the Beta distribution to determine the overall reputation of the camera node, i.e. how trustworthy/untrustworthy it is.

For a given time instance t, a camera node receives instantaneous positive and

negative ratings based on features observations as follows:

$$s_t = \frac{|V_t|}{|F|} \in [0,1]$$
$$\bar{s_t} = \frac{|U_t|}{|F|} \in [0,1] \tag{7.3}$$

where

$$(s + \bar{s}) \in [0,1]$$
$$|F_t| = \text{total number of features that should appear at } t$$
$$|V_t| = \text{number of features observed at time } t$$
$$|U_t| = \text{number of features not observed at time } t$$

It is important to note that $U_t + V_t \leq F_t$. We will go into detail on how U_t and V_t are calculated in the next section. The pair, $(s_t, \bar{s_t})$, gives an instantaneous snapshot of the trustworthiness/untrustworthiness of the camera by using only the features observations from the current time instance and looking at features observations that positively correlate as expected along with feature observations which deviate from what is expected.

We want to incorporate the instantaneous ratings with ratings from previous time instances in order to make the system more robust to noise. However, we want the current time instance to have more influence on the ratings. To do this, we aggregate ratings over time with a weighting variable as follows:

$$\begin{aligned} s_t^a &= \lambda s_{t-1}^a + s_t \\ \bar{s}_t^a &= \lambda \bar{s}_{t-1}^a + \bar{s}_t \end{aligned} \tag{7.4}$$

where s_t^a and $\bar{s}_t^a$ are the aggregated positive and negative ratings respectively and λ is a weighting factor which determines how much ratings from previous time instances will influence the aggregated rating. This gives the aggregated positive and negative ratings for the camera which look over a longer period of time, making the measure more robust to noise.

A single reputation for the camera is then assigned based on a combination of the aggregated positive and negative ratings. This reputation determines how trustworthy the data from the camera is and how likely it is that the camera has been attacked. The reputation is assigned on a scale of $[0, 1]$, where 0 is fully untrustworthy and 1 is fully trustworthy. We use the expected value of the Beta distribution to determine overall reputation as follows:

$$r^t = \frac{s_t^a + 1}{s_t^a + \bar{s}_t^a + 2} \tag{7.5}$$

The Beta distribution is well suited to our problem and gives a statistical backing to the reputation assignment, while allowing it to still be robust and flexible.

Once the overall reputation of the camera node falls too low, the camera node is

deemed untrustworthy. Appropriate steps can then be taken. Depending on what the application of the system is and which algorithms are running, these steps could be as simple as notifying an appropriate entity that the camera is untrustworthy and it might need to be checked out or more complicated, as appropriately weighting measurements from the cameras in automatic calculations such as 3D reconstruction or tracking estimates.

7.3 Camera Methods

In the previous section we discussed generally how the reputation is assigned to a camera node based on features. In this section, we will go into details about the different types of features to use and how to relate the observations, i.e. space-time correlations, given the three differing types of camera configurations in the network. The three steps we cover for each type of feature are:

1. **Feature Detection:**What the desired features are, what they represent in the scene, and how to detect them in an image.

2. **Feature Correlation:** How to correlate the features and what camera setup is needed. The correlation can be an intra-camera step, where the features in one frame of a camera are compared against features from previous frames of that same camera, or it can be an inter-camera step or where features from one camera are compared against features from another camera.

3. **Instantaneous Rating:**How the features observations and correlations are used to assign the instantaneous trustworthy and untrustworthy ratings to each camera. From this, overall reputation follows as described in the previous section.

In this section we also focus on the first step in the reputation assignment, the instantaneous ratings assignment of equation 7.3. This is the step directly affected by use of different feature observations. Once the ratings are assigned, the reputation follows directly from equations 7.4 and 7.5.

We discuss what type of attack each feature can be used to detect in the reputation system and the limits in the number of compromised nodes that can be detected.

7.3.1 Static Features

Static features are features that are permanent in the scene and we expect to see there consistently over time. In order to use static features for reputation assignment, first a set of robust static scene features, F, must be initially chosen. These features can be learned automatically using methods from the computer vision community, such as [55, 14, 90, 86, 45] to detect local features in a camera's field of view. The features can also be chosen manually by picking out landmarks in the scene, such as buildings or areas of interest, and building a detector/descriptor of each these.

Once the set of robust features, F, is determined, at each time instance, t, the feature detector is run to see whether these features exist in the image and at what

location. At each time instance, we obtain:

$$\begin{aligned} O_t \subseteq F = \text{subset features of F observed at time} \\ \bar{O}_t \subseteq F = \text{subset of features of F not observed} \end{aligned} \tag{7.6}$$

When looking at equation 7.3, the naive approach would be to make $V_t = O_t$ and $U_t = \bar{O}_t$. However, since the camera is observing a dynamic environment, it could be that a feature is counted in $\bar{O}_t$ due to an object in the foreground occluding it, not because the feature is not actually still present in the scene. This is shown in Figure **??**. Conversely, we do not want to credit the camera if a feature appears on a foreground object and not in the scene as we are unsure if that feature is actually still present in the scene. It is important that we only credit or discredit the reputation of the camera when features we expect to see are respectively either in the scene or no longer present in the scene. We want the minimize the effect of environmental noise, due to the scene dynamics, on the reputation of the camera.

To make our reputation robust to this environmental noise, we take occlusions into account and divide the image into foreground and background. The parameter V_t and U_t are then as follows:

$$\begin{aligned} V_t \subseteq O_t = \text{features of } F \text{ observed in background} \\ U_t \subseteq \bar{O}_t = \text{features of } F \text{ not observed in background} \end{aligned} \tag{7.7}$$

We do not look for features in areas considered as foreground in the image to make the

instantaneous ratings more robust as the feature might still be present, but occluded or not present, but appearing on foreground objects. While the first of these will occur more often, it is important to consider both situations as they will incorrectly influence the reputation of the camera if not addressed. Using the only the background for the reputation makes the result more robust to noise and the dynamic environment. This gives a more accurate description of which cameras are actually untrustworthy and have been attacked as opposed to just picking up on environmental noise.

We have explained the static feature method given a single camera, but it can be easily generalized to the multiple cameras with overlapping fields of view setup. With the overlapping cameras F now becomes a common set of static features both cameras see and the reputation is now a pairwise reputation. Given two camera C_i and C_j then the static features are F_{ij} and the pairwise reputation is judged by each camera, R_{ij} and R_{ji}. To determine a single camera's reputation, then the following is done:

$$\begin{aligned} R_i &= mean(\textstyle\sum_j R_{ij}) \text{ where} i \neq j \text{for } j \in 1, ..., n \\ R_j &= mean(\textstyle\sum_i R_{ji}) \text{ where} j \neq i \text{for } i \in 1, ..., n \end{aligned} \tag{7.8}$$

The multiple camera case does not give any more information than doing the single camera case, thus for static features, we stick to single cameras.

For the single camera case, the reputation assignment based on static features must be done centrally, by the base station or another trusted entity since only the

single camera can report on its data for its reputation. If the reputation is assigned on the camera's end of the network and then reported back to the base station, a compromised camera could always report itself with a trustworthy reputation even if it is compromised. Instead, having the base station either analyze the images the camera is sending back or the consistency of the feature observations the camera is sending, and then applying the reputation scheme to this information, is more robust to attacks. A much more savvy adversary is needed in order to trick the reputation scheme for a single camera if reputation is done at the base station. The adversary would have to accurately imitate observations of the static features that the base station has chosen or, they would have to put the camera in a feedback loop.

Using the reputation system on static features within a single camera, it can be determined if an non-invasive attack has happened. However, an invasive attack, such as replay loops or synchronization changes, will not be detected. An invasive attack where faulty data is injection, but is not a replay loop, can be detected if the observed features in the faulty data vary enough from what is expected in F.

7.3.2 Overlapping Motion Features

In addition to static features, overlapping motion features can also be used. We define overlapping motion features as motion of dynamic objects moving through the scene viewed by two or more cameras at any time. Thus, the overlapping motion features can only be used with cameras which have overlapping fields of view. In

addition, to use this method there are two assumptions that must be present in the system. First, the cameras must be synchronized within some error, and time-stamps should be given on the images. Second, the cameras are calibrated to a world coordinate frame.

Given some set of cameras $C_1, ..., C_n$, moving objects are segmented out of the images using background subtraction. This leave us with a set of foreground objects, O_i, for each camera C_i. We represent the objects by a point. The point that is chosen can vary depending on how much knowledge there is of the camera setup relative to the scene. If there is a known ground plane in the scene or known scene configuration, we can choose either the point on the object that should lie on the ground plane, for example point where the feet touch the ground for a human. Otherwise, the centroid of the object can be used as the point. Depending on what type of point is chosen will just effect how much error needs to be allowed in the measurements when doing correlation. For each adjacent time period of, $t_0 - t_n$, the points on the objects are tracked and then correlated. The tracking can be done using any sort of multi-target data association algorithm. This leaves us with object image tracks to use as features for doing correlation.

For each pair of overlapping cameras, C_i and C_j we then have:

$$\begin{aligned} T_i &= \text{the tracks in time } t_0 - t_n \text{ in camera} C_i \\ T_j &= \text{the tracks in time } t_0 - t_n \text{ in camera} C_j \end{aligned} \tag{7.9}$$

Since C_i and C_j are different cameras, they do not have complete overlap in their fields of view, thus T_i might not be equal to T_j as motion could be happening in the non-overlapping portion of the cameras. If there is a known ground plane in the scene that the cameras are oriented to, then the location on that ground plane of the object can determine if the object is in the overlapping portion of the cameras. If there is no common ground plane, we can use a more general approach and just take the maximum number of tracks between the two cameras as the number of motions that should match up:

$$|F| = max(|T_i|, |T_j|) \tag{7.10}$$

Without loss of generality, assume $|T_i| \geq |T_j|$, then each track in T_i is mapped to the best single track in T_j. For a given point, $x_i \in T_i$, and the corresponding point $x_j \in T_j$, then the reprojection error is:

$$d_j^2 + d_i^2 = \frac{(x_j^T \hat{T} R x_i)^2}{||\hat{e}_3 \hat{T} R x_i||} + \frac{(x_i^T \hat{T} R x_j)^2}{||\hat{e}_3 \hat{T} R x_j||}$$

where $e_3 = [0, 0, 0, 1]^T \in R^3$ and R and T are the rotation and translation of the one camera coordinate system into the other. This gives us a measurement on how correlated the two points are. For more details reprojection error please refer [77, 96]

The mean reprojection error then between tracks is then used as the measure

of how well those two tracks correlate. We choose the correlations which have the smallest mean reprojection error within some bounds. If the mean reprojection error falls outside this bound, we say the tracks do not match up. Thus, we are left with:

$$\begin{aligned} T_m = &\text{ number of tracks pairings within the error bound} \\ T_{\bar{m}} = &\text{ number of track pairings that don't match} \end{aligned} \tag{7.11}$$

the:

$$\begin{aligned} V_t &= T_m \\ U_t &= T_{\bar{m}} \end{aligned} \tag{7.12}$$

and the instantaneous rating can be assigned, but it is a pairwise instantaneous rating R_{ij}. Thus, there is an additional step that must be taken to find the reputation of each camera:

$$R_i = mean(\sum_j R_{ij}) \text{ where} i \neq j \tag{7.13}$$

This method using overlapping motion can be done one of two ways. It can be completely carried out at the base station by either the cameras relying their images or motion features. Or the pairwise reputations can be done in a distributed fashion, with each camera keeping track of pairwise reputations for those cameras which it

has overlap with. These distributed pairwise reputations can be fed back to the base station and the summation from equation 7.13 can be carried out.

Using overlapping motion features, invasive attacks can be detected, such as replay loops, where the static feature would still match up, but the motion features would not be likely to match up. Thus, the use of overlapping motion features is

7.3.3 Non-overlapping Motion Features

Similar to overlapping motion features, non-overlapping motion features depends on dynamic objects in the scene. However, here we assume there is no overlap in the motion, i.e., the dynamic objects are not seen at the same time by the same cameras. Yet, there is some correlation in one they leave the field of view of one camera and enter the field of view of another camera. This happens when cameras do not have any overlap in their fields of view or even for cameras that have some overlap, but for the areas where this overlap is no longer present.

If there is some correlation of motion between cameras, a distribution on when moving objects exit the field of view of one camera and enter the field of view of another can be found as shown in where it was used for tracking. We expand upon this idea to not only use time of departure and arrival between fields of view, but also optical flow of the object, as delta between the time of departure and arrival will have some correlation with flow. The distribution that applies to the non-overlapping motion can be learned by collecting the state information and then building a distri-

bution in the necessary manner.

Once this distribution is known, then given two camera, C_i and C_j with a distribution P_{ij} on the motion between their fields of view, we can use this to build a pairwise reputation. Given a moving object in the scene with state $x = k$, then the we can determine what what $P_{ij}(x = k)$ is. If there are N objects in the scene, and $n \in N$ have states lying in P and $\bar{n}$ do not lie in the distribution P, then the reputation is assigned as follows:

$$\begin{aligned} V_t &= \frac{n}{N} \\ U_t &= \frac{|\bar{n}|}{N} \end{aligned} \tag{7.14}$$

then the pairwise reputation are used to determine each camera's reputation as in 7.13

By modeling the motion into the feature observations, we can use the correspondence for between what is observed and what our model is, to base the reputation on. For example, if we have a model of the exit time of objects from one camera's field of view to the entrance time into the other camera's field of view, then we can use this to compare against when we notice moving objects in the scene. In they build a model of how the motion correlates... in our paper we use this kind of an idea, but extend the model to include optical flow and build the reputation around this.

This method allows us to detect invasive attacks even when the cameras have no overlap in their fields of view. Thus, this allows a broader array of configurations for

our camera network setup while still having security measures.

7.4 Experiments

We demonstrate the image-based reputation method using the different features on varying setups.

7.5 Static Features

Our first camera network consists of four cameras which have no overlap and no correlation in motion between them. Since these cameras have no overlap, we performed attacks on one of the cameras and looked at how static features are used to form reputation.

The frame rate on this experiment was approximately 1 frame per second. Since there is no overlap, the image-based reputation can be formed on only static features within each camera's field of view. In order to obtain the set of static features F, we took image data from 10 minutes of video and ran a Harris corner static feature detector to learn what a robust set of static features is for F. To detect foreground objects we use a version of the background subtraction method of [100]which is based on Gaussian Mixture Models on pixel values. A value of 0.7 is used for λ in this experiment.

Here we ran three different types of attacks. First, we ran an attack where a

camera's field of view is obscured by something covering it. For this attack, we covered the field of view with a plastic bag we tied into a balloon and floated in front of the camera. Images from this attack can be seen in 7.3. The reputation results are shown in 7.4.

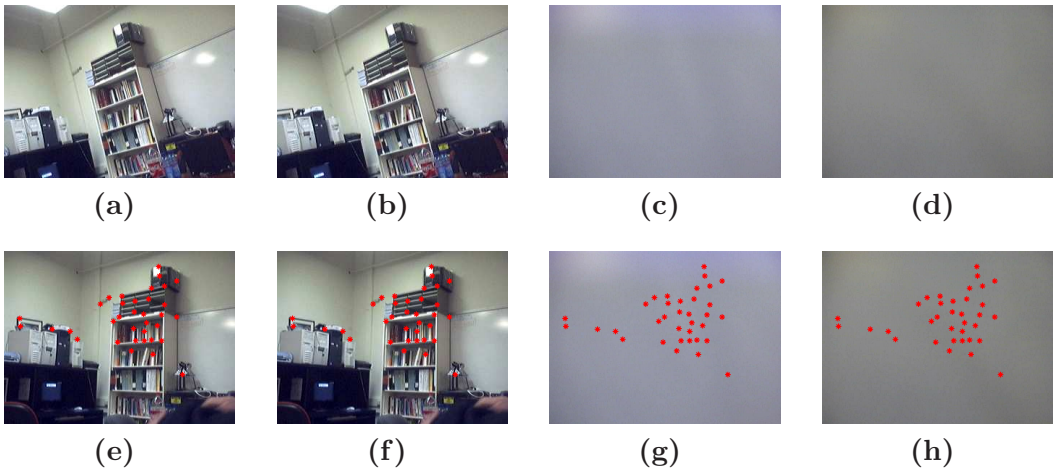


Figure 7.3: Lens Covering and static features: Images from the attack where this camera's lens is obscured by a bag. (Top Row) Subfigures (a) and (b) shows the field of view of the camera before it is obscured by the bag in subfigures (c) and (d). (Bottom Row) Subfigures (e) and (f) show that all the features from the static feature set are detected before the attack and (g) and (h) show the static features in red that should be detected and the actual features that are detected in green which do not correspond to the static feature set.

Second, we ran an attack where one of the cameras is physically moved.

Third, we ran an attack where the lense of the camera is blinded by a laser. We used a laser pointer and aimed it at the ccd of the camera to blind the lense. The images from this attack are shown in 7.5. The reputation from this attack is shown

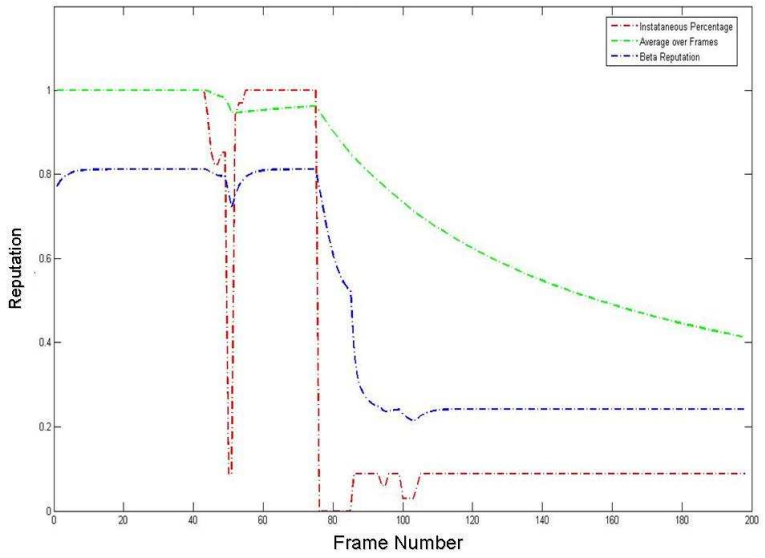


Figure 7.4: Lens Covering of a Single Camera:The graph shows the difference between using instantaneous percentages of features seen, an average of the number of features points seen, and using a reputation based on static features. The lens of the camera is covered by a bag at frame 75. The instantaneous percentages are very sensitive to occluding objects hiding features while the average is not sensitive enough to changes in the scene as it takes a while to fall off after the lens has been covered. The reputation system provides a good balance of sensitivity and reactivity.

in 7.8.

As can be the image-based reputation method does detect these attacks and is only triggered by these attacks, not by the noise in the environment. We show the results from only one camera to eliminate redundancy as each camera is treated individually for static features. We compare this against the results if just the instantaneous percentage of visible features is used for the reputation and against the average,over

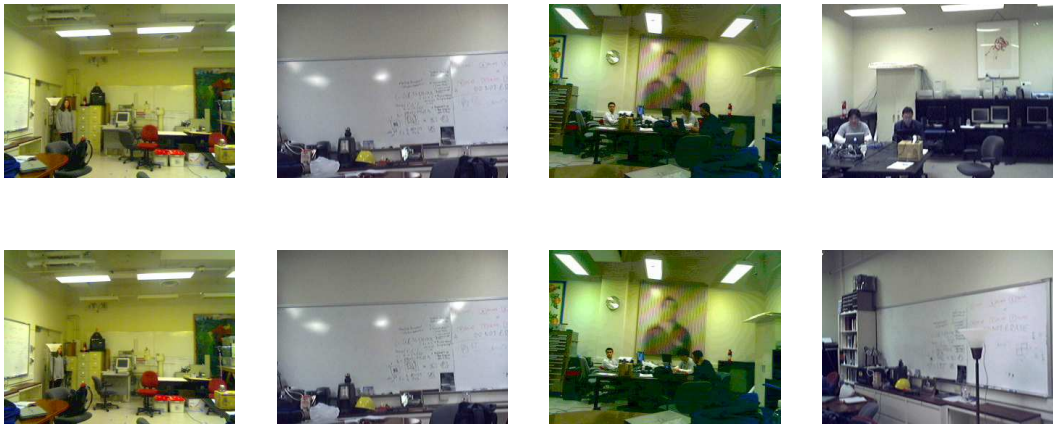


Figure 7.5: Moving a Camera Attack. (Top Row) Original views from the fields of view of the cameras. (Bottom Row) Views from the cameras with the third camera being moved.

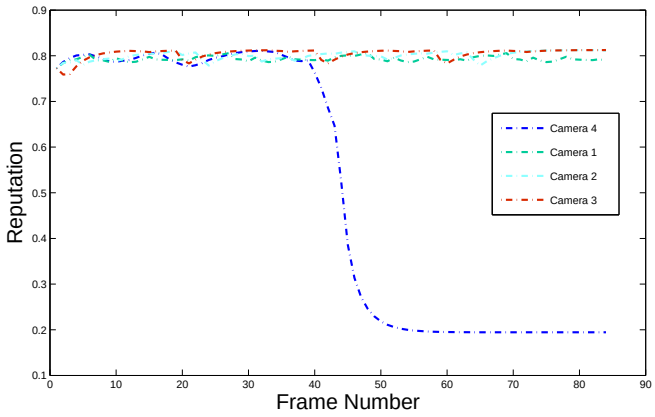


Figure 7.6: Moving of a Single Camera:The graph shows the difference between the reputation, based on static features, of the cameras that are not moved and the camera that is moved. Camera 4 if moved at frame 42 so that it's field of view points in a different direction.

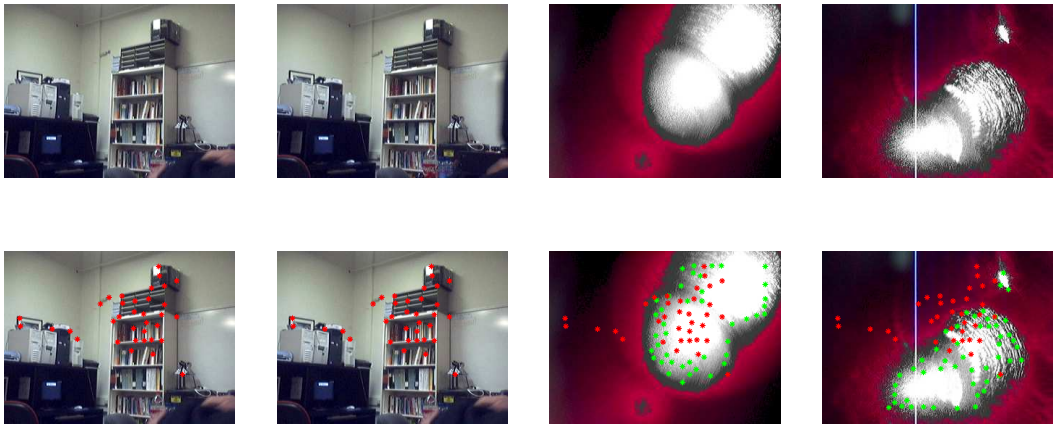


Figure 7.7: Blinding Camera with Laser Attack. (Top Row) shows images from the attack. (Bottom Row) Shows the static features. The red features are the set of robust static features. The green features are the sets of detected features for that image. In the first two images in this row, the robust static features overlap with the detected features, thus no green features appear. In the left two images, the detected features do not overlap with the robust static features.

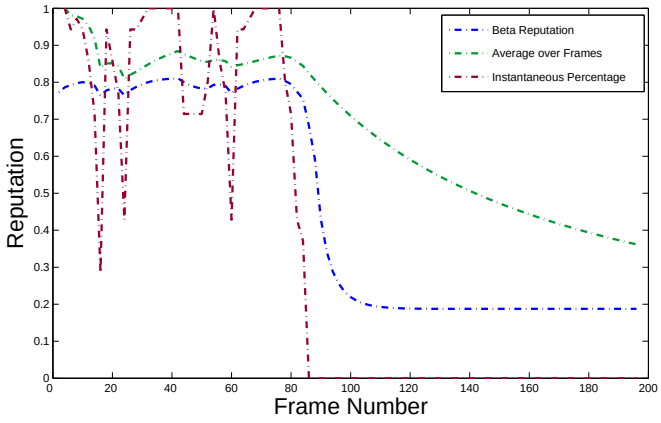


Figure 7.8: Lens Blinding of a Single Camera:The graph shows the difference between using instantaneous percentages of features seen, an average of the number of features points seen, and using a reputation based on static features. The lens of the camera is blinded by a laser at frame 84 and after. The instantaneous percentages are very sensitive to occluding objects hiding features while the average is not sensitive enough to changes in the scene as it takes a while to fall off after the lens has been blinded. The reputation system provides a good balance of sensitivity and reactivity.

frames, of visible features as the reputation. It is shown that the instantaneous percentage is very sensitive to features getting occluded. In frame 50 many features are occluded, due to environmental noise, and this causes the percentage of visible features to dip very low. If the owner of the camera system was to be notified whenever a camera's reputation got too low, as a signal of an attack, this method of reputation would cause many false alarms. The use of average percentages across frames is not as beneficial as our propose reputation system as it takes a long time to decay once the camera is covered by the bag. Thus, the owner of the camera network would not

be notified later in the process than when our reputation system would notify that the camera reputation is too low. The noise of the environment does not cause a camera's reputation to become untrustworthy, as the intantaneous feature rating causes and the reputation responds more immediately to the attack than another approach just taking average number of features. Thus, we show that the method is robust to noise, but sensitive to the attacks.

7.5.1 Overlapping Cameras and Overlapping Motion Features

We also tested the reputation system on a multi-camera setup that where the four cameras all overlapped in a portion of their fields of view with one another. The four cameras placed around a 20 x 30 room for this experiment. The cameras were synchronized to a global clock, so images from all cameras were taken at the same time instance. Images of objects moving through the scene were taken. We spliced together parts of two sequences, defined as sequence 1 and sequence 2, from camera 1 and used this to imitate an attack where the adversary sets the camera in a playback loop. We compared this against the complete sequence 1 from the rest of the cameras. The sequence is 99 frames long, with the playback loop inserted at frame 51. The scene consists of an empty area with a person walking through it at frame 43 till frame 99. The results are shown in Figure 7.12. It can be seen that the reputation based on comparing features seen in the frames of camera 1 does not look

significantly different than the reputations of the other cameras based on this same measure. However, the reputations using foreground object correspondences between multiple cameras shows that an attack is happening. By looking at the reputation on each pair of cameras, it can be seen that the reputation on any pair involving camera 1 below the acceptable threshold for a good camera pair. By looking at the overlap of the cameras pairs that fall below the acceptable threshold, it can be determined that camera 1 is the camera that has been attacked.

If we use the static feature reputation along with the reputation on pairs of cameras due to foreground objects, it can be determined that a playback attack is likely the cause. In order for the static reputation to remain good while the pairwise reputations involving that camera fall below the acceptable threshold, the camera would have to be observing the same background seen, yet seeing different motions of the foreground objects in the scene. While this could possibly be caused by another attack, for instance the camera getting moved to an area where the same type of features will appear in the image at the same locations, yet the object motion would differ, this is highly unlikely. It would by very difficult to find a way to place the camera in a different area yet have the background features remain exactly in the same place. It is more probable that the camera was put in a playback loop as an attack on the network, just replaying previous frames captured from the camera over again. This would cause images from the camera to still show the same feature points from the seen, yet show different motions of the foreground objects.

Figure 7.9: Original views from the cameras with a moving object in the scene

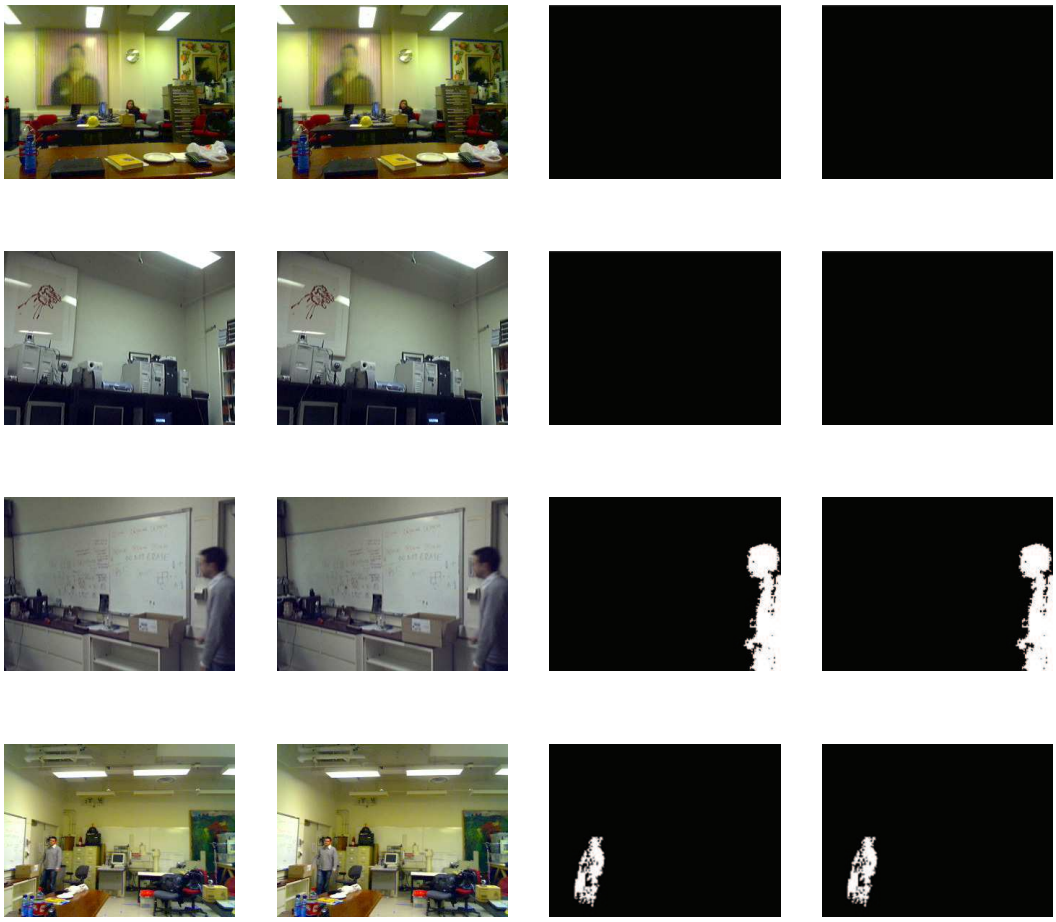


Figure 7.10: Sequence of moving object in a pair of cameras

Figure 7.11: Replay Attack. (Top Row) The camera where the replay attack happens. The second two frames show the views when the replay attack is occuring. (Bottom Row) Another camera in the network that is trustworthy and has not been attacked. As can be seen, the two cameras should both not see any moving objects if they are trustworthy, but the top camera shows moving objects due to the replay attack.

7.5.2 Non-Overlapping Cameras and Non-Overlapping Motion Features

Each camera network used Panasonic KX-HCM280A IP cameras and we used their network capabilities to obtain data. These are commercial cameras and of a quality that is becoming more common in surveillance camera network settings. The cameras have remote 350 degree pan and 220 degree tilt and a 21x optical zoom. Our cameras were synchronized with one another and each frame was time stamped.

On our third setup we looked at two non-overlapping cameras which had a correlation of motion between when moving objects exited the field of view of one camera and entered the field of view of the second camera. In this camera network, Panasonic KX-HCM280A IP cameras and were used to obtain data. These are commercial cameras and of a quality that is becoming more common in surveillance camera network settings. The cameras have remote 350 degree pan and 220 degree tilt and a 21x optical zoom. Our cameras were synchronized with one another and each frame was time stamped with a rate of 5-12 frames per second over the network.

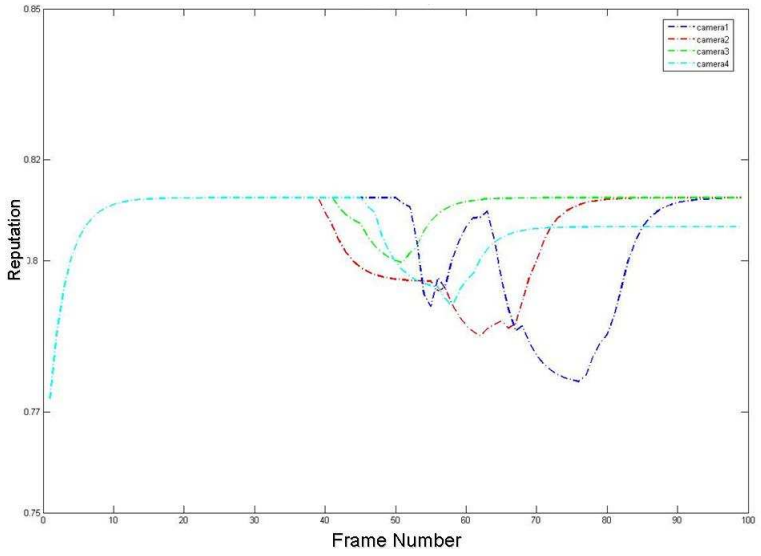
For this setup, we took twenty training sequences between a camera pair we define as (C_i, C_j) and manually determined the entrance/exit points of objects between the cameras. We then created a histogram of the transition times of objects between exit from one field of view, either C_i or C_j, to entrance in to the next field of view, respectively C_j or C_i. We fitted a gaussian distribution to this histogram to use as our model of transition times between the cameras, as shown in Figure 7.5.2, and set

a good transition to be any transition time, k, such that $P(x = k) > 0.1$. For the replay attack, we took an earlier sequence captured from camera C_i and replayed this in a loop starting at frame 100 while camera C_j was showing the true data it was capturing. The resulting reputation for the camera pair is shown in Figure 7.5.2.

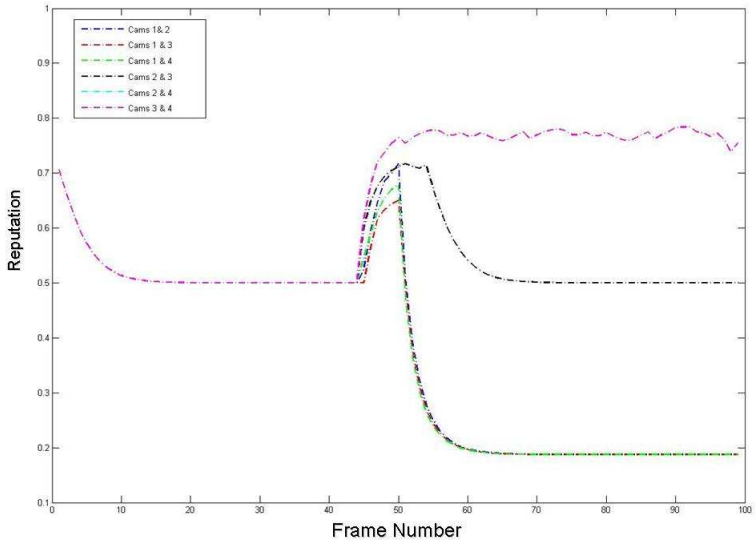
7.6 Conclusion

In this chapter we have presented an image-based reputation system that can be used for detecting attacks on the camera nodes themselves. We have shown how varying features from images can be used to detect different types of active attacks and have shown their tradeoffs. While the use of static features is more robust in detecting certain types of attacks, as the reputation can be calculated off these features at every time instance, the reputation on these features can only be used to detect non-invasive attacks. On the other hand, the use of dynamic features to detect attacks can be used to detect invasive attacks, but will is not as robust as static features as the reputation is dependent on moving objects in the scene being present and correlation between these object.

This method of detecting attacks on the cameras in the network provides additionally security for the network that compliments previous methods for protecting data in transmission and storage. By presenting this method to detect attacks on the cameras before their information is transmitted, we have provided a way to check on the data integrity of the cameras and help further the security of the camera network.



(a)



(b)

Figure 7.12: Replay Attack: These graphs show the reputations of the cameras during a playback loop for the replay attack which begins at frame 50 in camera 1. the graph shown in (a) shows the static feature reputation for each camera while the graph in (b) shows the pairwise reputations. Only the pairwise reputations show any sign of attack while the static reputations for all the cameras look similar.

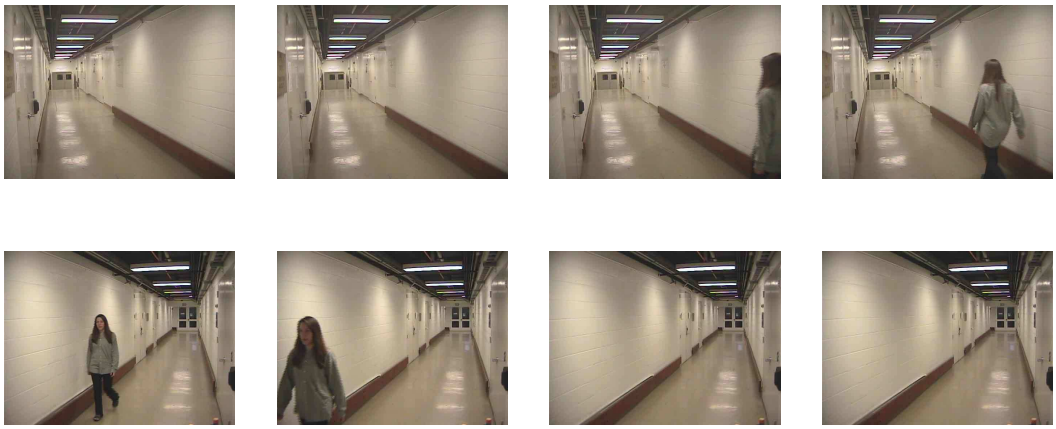


Figure 7.13: Nonoverlapping Cameras

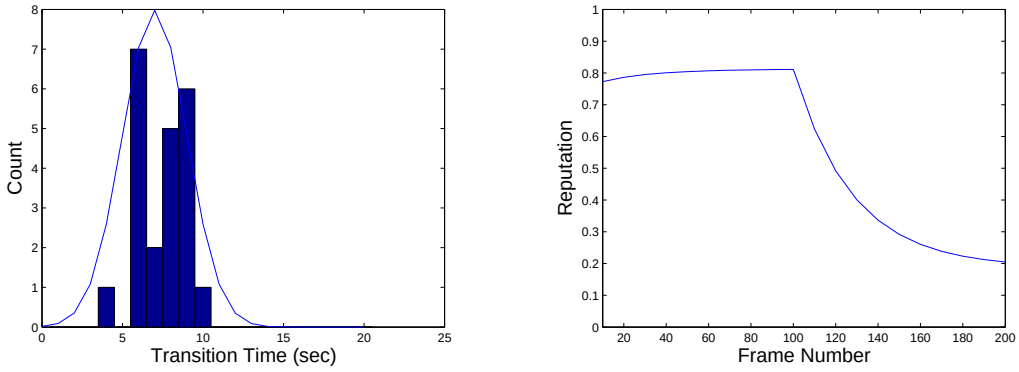


Figure 7.14: Replay Attack on Non-Overlapping Cameras. (Left) The model of the transition time distribution. The bar graph are the results from the training sequences and the gaussian curve is fitted to this data. (Right) The pairwise reputation R_{ij} for when the given replay attack happens on camera C_i.

Chapter 8

Introduction to Privacy and Camera Networks

> *Civilization is the progress toward a society of privacy. The savage's whole existence is public, ruled by the laws of his tribe. Civilization is the process of setting man free from men.*
>
> Ayn Rand (1905 - 1982), The Fountainhead (1943)

In the past decade, particularly since the events of September 11, 2001, the placement of camera networks in public spaces has increased both in the U.S. and internationally. For example, closed circuit televisions systems (CCTV) have grown in recent years in Britain, with over 4 million cameras surveying the public [50]. In major U.S. cities, such as Chicago and New York, camera networks are quickly gaining ground with hundreds of cameras owned and operated by public agencies and thousands more by private enterprises [9, 62, 4, 58]. Current applications for these networks include crime detection and traffic monitoring [36, 49, 88, 19].

The use of these camera networks in public spaces has demonstrated benefits. For example, the footage from the London tube stations were used to help identify the attackers from the 2005 bombings [85, 69]. However, camera networks in public spaces also raise the issue of privacy for the individuals being monitored.

The importance of considering the privacy issues with cameras in public spaces is necessary in order to gain the benefits of the technology while limiting the harm to individuals and maintaining the values of society. Policy-makers in many places, such as Britain [33], Canada [63], and Sydney, Australia[1, 15] have already attempted to address privacy issues by making guidelines as to where the camera network will be placed, requiring notices of recording, and identifying who has the right to see the image data. Additionally, policy researchers have looked at these issues of surveillance and privacy stressing the importance of protecting privacy rather than assuming a de facto trade off of public privacy in the name of increased security [89, 31, 23]. For example, the U.S. Constitution Project Guidelines for public surveillance systems state that "A [Public Visual Surveillance] system should be designed and operated so that the privacy intrusion it creates is no greater than absolutely necessary to achieve the systems goals" [89].

Despite the policy guidelines and the discussion generated by policy-makers, few actions have been undertaken to determine public perceptions of surveillance systems and what their privacy expectations are with regard to surveillance systems. This sort of analysis is necessary so that these camera networks can in fact be designed

and operated with as little intrusion as possible to an observed individual's privacy. The level of detail camera images capture is the obvious concern, but the need to understand public perceptions of surveillance systems with regard to privacy becomes particularly poignant when we recognize concerns beyond this. The remote or hidden nature of the cameras in public spaces and the inability in some cases for the camera networks to be avoided by the public, and the ease in which image data can be transferred without accountability for following privacy protection guidelines. These are some of the aspects which point towards the need to understand public privacy concerns with regard to camera networks.

To address this need, we conducted a study to explore *visual privacy*. We define visual privacy as being the relationship between the collection and dissemination of visual information and the public expectations of the collection and use of this visual information. We see visual privacy as a subset of the much broader topic of information privacy. In this paper we look at the visual privacy expectations a group of individuals being observed by camera network have. We explore these expectations in order to understand how they can inform the design of the surveillance systems to better preserve public privacy. Meeting the visual privacy needs of these observed individuals poses a interesting problem, namely, what happens when participants do not have control over the technology, but must interact within the space? These participants are not users in the sense that they directly interact with the technology, but they are still stakeholders in the system as information about them is getting

used. What are these participants' expectations, and how can a system be designed to incorporate these expectations when the participants do not directly interact with the technology?

8.1 Related Work for Visual Privacy

In examining visual privacy, there have been a number of works that propose privacy protection by obscuring an observed individual in different ways. A group of techniques have focused on obscuring the face of observed individuals. In an approach by Kitahara and Newton the de-identification of faces is done using previous knowledge of the person to anonymize the face [39, 61]. Another technique to de-identify faces utilizes a combination of RFID and image processing [92]. Yet another approach is presented in [79]which uses visual markers that observed individuals wear and then use image processing in real-time to obscure faces.

Some techniques have focused on hiding the whole human by just showing a solid silhouette. An example of this is shown in IBM's People Vision system, where people that are moving and in the foreground are obscured using robust foreground/background detection and human silhouette models [97, 17]. This type of approach shows no detail of the moving observed individuals, but does show large scale motion such as walking (gait) or jumping. Similar to this, methods of blurring the whole image hide fine details of the observed individuals in the scene, but still show large detail and show large motion as well. An example of this sort of approach is done by 3VR [6].

While these differing approaches for de-identifying faces and humans have been proposed, these works do not examine what visual privacy actually means for observed individuals and whether it is actually being preserved by these approaches. Some research addressing this gap between observed individuals' expectations and filtering measures includes a study looking at blur filtration in home-based video conferencing systems [60]. This study states that only bluring at certain levels is actually deemed privacy protective by subjects and that the level necessary is dependent on the activity taking place. Additionally, a study looking at a group of women and their views on visual privacy in relation to who is watching the video is presented in [40]. In this study, different types of abstraction methods are applied to a single image and the image is then shown to the subjects. The women are then asked to rank how they feel about privacy given their relation, on sliding scale of familiar to unfamiliar, to the people possibly viewing these images.

Our study is is similar to these last two efforts in that we examine visual privacy from the observed individual's perspective and are interested in the interplay between expectations and technical measures. However, our focus is on surveillance in public spaces and the observed individuals' expectations in this space. Unlike video conferencing settings, with public video surveillance the observed individual has no direct interaction with the technology, yet remains a stakeholder.

In addition, we expand upon the other works by actually placing our subjects under surveillance. Thus, when reporting how they feel about privacy, subjects had

actual footage of themselves to judge. This brought the concept of visual privacy into the personal realm. The subjects were also shown video of themselves, not just still frames, so they received and reacted to both the spatial and temporal information video conveys.

Another difference with our work is that we examine multiple aspects and their effects on visual privacy expectations. Specifically, we look three different aspects: 1) activities being performed 2) place where the surveillance is and 3) identifying factors of race, gender, and face.

Lastly, our study looks at filtering measures and how they might be used as embedded technology to meet observed individuals' privacy expectations. We present four different filtering measures, that are possible using only visual data and no prior knowledge of the observed individuals, and examine if they uphold visual privacy expectations. We take cues from proposed privacy protection techniques in the related works and test these types of obscuring ideas in addition to some others, which are further discussed in the Study Design section.

8.2 Contributions of Our Work

Our study is is similar to these last two efforts in that we examine visual privacy from the observed individual's perspective and are interested in the interplay between expectations and technical measures. However, our focus is on surveillance in public spaces and the observed individuals' expectations in this space. Unlike video confer-

encing settings, with public video surveillance the observed individual has no direct interaction with the technology, yet remains a stakeholder.

In addition, we expand upon the other works by actually placing our subjects under surveillance. Thus, when reporting how they feel about privacy, subjects had actual footage of themselves to judge. This brought the concept of visual privacy into the personal realm. The subjects were also shown video of themselves, not just still frames, so they received and reacted to both the spatial and temporal information video conveys.

Another difference with our work is that we examine multiple aspects and their effects on visual privacy expectations. Specifically, we look three different aspects: 1) activities being performed 2) place where the surveillance is and 3) identifying factors of race, gender, and face.

Lastly, our study looks at filtering measures and how they might be used as embedded technology to meet observed individuals' privacy expectations. We present four different filtering measures, that are possible using only visual data and no prior knowledge of the observed individuals, and examine if they uphold visual privacy expectations. We take cues from proposed privacy protection techniques in the related works and test these types of obscuring ideas in addition to some others, which are further discussed in the Study Design section.

To address this need, we conducted a study to explore *visual privacy*. We define visual privacy as being the relationship between the collection and dissemination

of visual information and the public expectations of the collection and use of this visual information. We see visual privacy as a subset of the much broader topic of information privacy. In this paper we look at the visual privacy expectations a group of individuals being observed by camera network have. We explore these expectations in order to understand how they can inform the design of the surveillance systems to better preserve public privacy. Meeting the visual privacy needs of these observed individuals poses a interesting problem, namely, what happens when participants do not have control over the technology, but must interact within the space? These participants are not users in the sense that they directly interact with the technology, but they are still stakeholders in the system as information about them is getting used. What are these participants' expectations, and how can a system be designed to incorporate these expectations when the participants do not directly interact with the technology?

In our study we actively surveil our subjects using a network of cameras and ask them about their response to visual privacy. With this study, we focus on two main issues. First, we look at privacy expectations in public spaces by analyzing our subjects' attitudes towards visual privacy in relation to three aspects: 1) activities being performed 2) place where the surveillance is and 3) identifying factors of race, gender, and face. Second, we look at how embedded design measures might be able to meet their expectations since they can not directly interact with the surveillance technology. We presented four different image-based filtering measures and analyzed

the subjects' responses to these measures in regards to privacy preservation.

Chapter 9

Subject Study on Privacy Expectations in Public Places

The quality of expectations determines the quality of our action.
André Godin (1817-1888)

9.1 Study Design

For this study, subjects participated individually in two different sessions. In the first session, subjects filled out a short survey and then were asked to walk a given route on a map and perform a set list of activities. During this walk, the subjects were unknowingly under surveillance in three different places and footage of them was recorded. When the subjects returned from their walk, they were asked to fill out a survey regarding different aspects of being out in public and questions about

surveillance.

The same subjects returned for the second session 3-7 days later. Ifn this session they were shown the videos that were recorded of them on their walk from the first session. The subjects were shown two types of videos: 1) the three original videos recorded of them while on their walk and 2) the three same videos with four different visual filtering methods applied. The subjects also completed another survey during this session regarding their views on video surveillance, privacy attitudes in general, and public behavior.

In this study, 24 subjects were recruited from a randomly selected pool of student and staff. Out of the 24 recruited, 21 subjects completed both parts of the study. Of these 21 participants, 11 were female and 10 male; their ages ranged from 18-26 years. All subjects had completed a high school or greater level of education.

9.1.1 Research Questions

In running this study, there are two main issues we focus on: 1) what are the privacy expectations observed individuals have in public spaces and 2) if the four proposed filtering measures uphold these expectations and how they can be used in embedded design.

In tackling these two issues, there are five research questions that are looked at:

Question 1: How do subjects feel performing different common activities in public?

Question 2: How do subjects feel about surveillance in different spaces?

Question 3: How do subjects feel about race, gender, and their face being able to be identified under surveillance?

Question 4: Do subjects feel their privacy is preserved by the four visual filters presented?

Question 5: Which filter, of the four presented, do subjects choose in order to preserve their privacy?

The first three questions address the issue of expectations of visual privacy in public places under surveillance. Each address a different aspect of what we think might affect visual privacy expectations. The last two questions address the visual filtering measures. These questions look at whether the filters are privacy preserving for observed individuals and what potential they have for embedded design solutions.

9.1.2 First Session: Scenario for Subject and Walk

Surveillance in public spaces captures differing activities and often times people being observed are unaware of the fact that their activities are being monitored. To emulate this in the experiment subjects came in for the first session and were told that they were doing a study on behavior in public places, but no mention of surveillance was made. Surveillance was specifically not mentioned in order not to influence how subjects acted in this first session. We wanted their reactions and participation to be as natural as possible. The first session had three main parts to it:

1. Pre-Walk Survey: Upon arrival, each subject filled out a brief survey about their demographics, including age and gender, and answer the questions for the baseline on performing activities in public.

2. Walk: Following the survey, subjects were given a map with a route they were to walk, a small shoulder bag with necessary items in it, and a list of activities they would perform along their walk. The route on the map was chosen such that it went through common public areas traveled by many people throughout the course of the day. Additionally, since the route was very common, the intention was that it would put the subjects at ease and they would act more natural in this common setting. The activities that they had to perform are activities that commonly take place in public spaces. The five activities the subjects performed, in order, and the locations are as follows:

   (a) Sit at a bus stop and read a magazine: This was the first activity on the walk and an activity that allowed the subject some choice. There were three magazines in the bag and the subject chose which one to read an article in.

   (b) Post fliers: Fliers for a local music show were provided in the bag and the subject had to post 2-3 fliers in a public square on common public notice boards.

(c) Sit on the steps and eat a snack: Subjects were asked to sit on the steps of a well-known campus building and choose a snack to eat from a bag we provided. This also was an activity had a choice involved.

(d) Take pictures of a landmark: In this activity, subjects were asked to take picture of a specified landmark.

(e) Ask an informational question: The last stop on the map led the subject to a common place visited on campus that has an information desk. Here the subject was given a question to ask the attendant at the desk.

3. Post-Walk Survey: After the walk, subjects returned and filled out a followup survey that had some questions regarding surveillance on it. Again, subjects were not told they had been under surveillance.

9.1.3 First Session: Surveillance Setup

In the first session of our study, while the subjects were on their walk, three of the five activities they performed were recorded by surveillance, specifically activities 1, 3, and 4. Not all activities were recorded as it was not necessary and due to physical limitations on where the cameras could be placed. The subjects still were asked to perform these two unrecorded activities as it provided the needed time for organizational purposes and provided the subject with a broader expanse of experience to comment on in the public space. Additionally, we did not want the subject to

feel that they were constantly under surveillance to reflect more realistic surveillance conditions.

For each recorded activity, a single camera was designated for recording. Three cameras were used, each placed at a location relative to the designated activity and each having a different type of viewing angle of the subjects. We used Panasonic KX-HCM280A IP cameras and used their network capabilities to obtain data. These are commercial cameras and of a quality that is becoming more common in surveillance settings. The cameras have remote 350 degree pan and 220 degree tilt and a 21x optical zoom. Between 4-12 frames per second were recorded over the network and remote navigation of the cameras was done with 1s lag time resulting. The activities recorded, as well as a demonstration of pan, tilt, and zoom capabilities of the cameras and the level of detail they can capture, are shown in Figure 9.1.

Figure 9.1: (Top Row) Frames from the three recorded activities for one subject. (Bottom Row) The same three activities with the camera zoomed in for a closer shot.

Each camera was actively monitored and maneuvered to obtain differing levels of detail on the subject and the activity, similar to what is often done in modern surveillance settings.

9.1.4 Second Session: Image Filtering Techniques

After obtaining the surveillance videos of the subjects, all the videos were put through through four different image filtering techniques. The four filtering techniques that were used are:

1. Facial Obscuring: In this method we cover the head of the subject with a opaque circle. This method imitates what is done in [79]. Since our subjects were not wearing prescribed visual markers because we wanted them to feel as normal as possible when performing the activities and because visual markers are not used in areas under surveillance with current systems, a combination of face detection and manual intervention was done to automatically hide the head. For frames where the face could not be detected, we manually located the head and placed an opaque circle over it.

2. Solid Silhouette: In this method, the subject is represented as a solid silhouette. The silhouette is all white and covered the whole person. Background subtraction was used to automatically form the silhouettes when the subject was moving and could be detected as foreground by automatic background

subtraction methods When the subject did not have enough motion, the solid silhouette was manually create.

3. Outlines: With this filter, the scene objects are outlined in white and everything else is black. This is a fully automatic method that is based on edge detection and curve formation techniques from computer vision.

4. Patch-Based Blur: This is an automatic filter which takes an area of the image and treats it as the same color and then blurs those areas across the image. The blur filter used in this study differs from what was tested in the media space work by [12, 60]. In that work, a Gaussian-based image blur was used, which simply blurs pixels in a neighboring radius together. The blur filter tested in this study is similar to what 3VR uses, in that it takes patches of pixels and forces them to be a single color and blur is then done across these single color patches. This way of filtering provides less detailed information than gaussian image filtering.

Figure 9.2 shows examples of these filtering techniques.

These four filtering techniques were chosen for several reasons. First, these techniques are only based on image features and do not depend on any prior knowledge of the subject or the scene. While not all of these methods are completely automatic at the moment, there is the potential for them to become so by using only image data. Since the observed individuals do not interact with the surveillance technology, this

study looks at visual privacy preservation measures that need no prior knowledge of the observed individuals nor any interaction with them. An example of a privacy measure to represent a person as a dot in the image with his/her name and age listed below was given in [40]. However, this would require prior knowledge of the subject and some interaction from the subject about their name and age. This is not possible in most surveillance setups and does not lend itself well to embedded design. By using filtering techniques based on image features alone, these techniques can be applied in any surveillance setting and do not require interaction.

Second, the four filters chosen are representative of many of the current techniques being proposed for use by companies or by the research community. As discussed in the Related Work section, facial obscuring, solid silhouettes, and blurring have all been proposed, in some form, as privacy solutions. However, the is very minimal work examining whether they are effective for observed individuals' privacy expectations and none done for privacy expectations in public places. Outline images have not been suggested for hiding identity, but outlines are often used in computer vision for picking out key shapes and objects. It is a natural form of image abstraction that might aid in visual privacy. Using these filtering techniques, we can analyze if the methods proposed by industry and research are actually effective at providing an observed individual with a sense of visual privacy and what direction(s) the development should continue down.

Third, the filtering techniques used in the study abstract a certain amount of information, but leave much of the scene data intact. Thus, if a camera operator is looking at the video with these filters on, they can still see the scene. With obscuring the head, only human heads are hidden from view from a camera operator. The rest of the scene in the video is entirely intact. In the solid silhouette, the humans in the scene become filled-in silhouettes, but everything else in the video scene remains intact. With the solid silhouette details, such as the face and the skin tone and what the person is wearing, cannot be seen. Also, small motion, like blinking, cannot be seen, but large overall motion of the humans, like walking or jumping, can be. With the outline filter, color is completely removed from the image and major shapes become outlines. Smaller, less prominent shapes do not appear and details, such as color and texture, are hidden from view. Prominent objects in the scene can still be detected as well as their motion. With the Patch-based blur, portions of the scene are blurred together. This hides details of the scene. For example, a face gets blurred together so you can no longer pick out the eyes and mouth or see the nose. Color and motion are still present in the scene.

9.1.5 Second Session: Follow-up and Survey

The subjects from the first session returned for their second session 3-7 days later. In the second session subjects were told that they had been under video surveillance in the first session. They then were asked to rank how comfortable they now felt

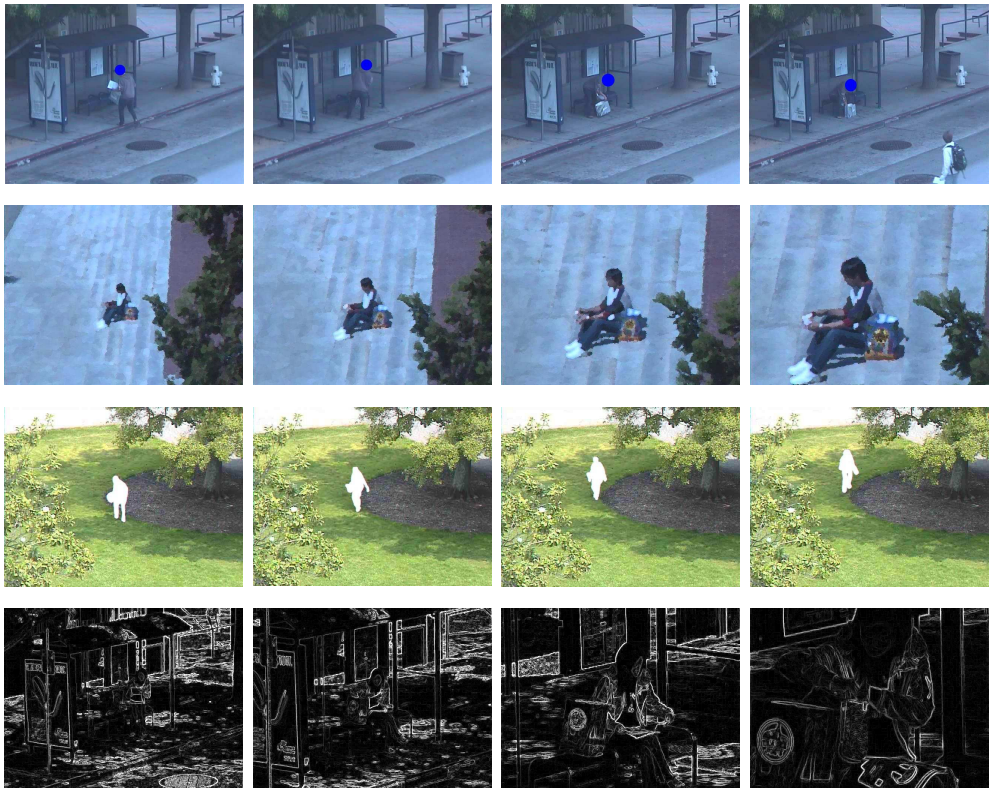


Figure 9.2: (First Row) The head obscuring filter at the bus stop activity. (Second Row) The patch-based blur shown on the stairs activity. (Third Row) the solid silhouette filter shown on the third activity at the stairs. (Fourth Row) The outline filter shown at the bus stop activity.

performing those five activities while under surveillance and were also asked which groups they would be comfortable with monitoring those activities.

They were next shown the unfiltered videos of themselves on their walk performing the three activities that were recorded. They were asked to state how they felt about being surveilled doing these activities. Following this, they were shown their videos under the visual filtering measures. The filtering measures were shown one at a time.

Immediately after a visual filtering measure was shown, and before moving on to the next filtering measure, the subjects were asked to rate the filter on how effective they felt it was at preserving their privacy. They were also asked to rank how comfortable they would feel performing the activities on the walk if that filter was in place. Once all visual filtering measures were shown, the subjects were then asked to rank the filters in order of most privacy preserving to least privacy preserving. It is important to note that no audio was present on any of the videos. As the focus is on the visual privacy aspect, only visual data was presented.

9.2 Results from Subject Participants

9.2.1 Results

The findings from this study are organized according to the two specific issues focused on: 1) what are the privacy expectations that observed individuals have in public spaces and 2) how do the four proposed filtering measures uphold these expectations and if they can be used in embedded design.

9.2.2 Visual Privacy Expectations

To discover what subjects' expectations of visual privacy are, we examined three different aspects: activity, place, and identifying factors. First, we looked at how comfortable subjects rated performing certain activities in public. We did this to see

if there was a trend with particular activities that subjects were sensitive to doing in public which might warrant privacy protection. In the first session the subjects were asked to rate each of ten activities commonly performed in public from uncomfortable to comfortable. The ten activities are:

1. Eating
2. Taking photographs
3. Playing sports
4. Making a call on a cell phone
5. Asking a stranger for information
6. Handing out or posting flyers
7. Wearing clothing that might call attention to yourself
8. Reading a newspaper, book, or magazine
9. Participating in a political activity
10. Talking to a friend

To determine whether there is a significant effect between certain activities and comfort of performing them in public, we did a within-subjects, one-way ANOVA test with activity being the independent variable, with ten levels to it each corresponding

to a specific activity. The dependent variable is the level of comfort. The results from performing the ANOVA are shown in Figure 9.3

As can be seen from the results, subjects are not as comfortable performing certain activities in public as they are with others. In particular, activities 6, 7, and 9 are more sensitive for subjects. These correspond to handing out posters or flyers, wearing clothing that might call attention to yourself, and participating in a political activity.

Next, we looked at how subjects felt about the use of surveillance in different settings. Ten options were presented of the types of places where surveillance might be found:

1. Privately owned store
2. Shopping mall
3. Bank
4. University campus or building
5. Public parks or plazas
6. Residential streets
7. Sports stadium
8. Public transit vehicles
9. Airports

10. Parking Garages

These ten places were chosen as they are representative of places where surveillance cameras are currently found. Again, we ran a within-subject, one-way ANOVA to see if expectations of where surveillance is appropriate varied among places. The independent variable is place and there are ten level each representing a specific place. The dependent variable is appropriateness of use of surveillance. The results from performing the ANOVA are shown in Figure 9.4.

From the results, it can be seen that the level of appropriateness subjects feel for using surveillance does vary based on place. In particular, subjects feel public places such as parks and plazas and residential streets are less appropriate for using surveillance in as opposed to privately owned spaces and enclosed spaces, such as stores, malls, banks, airports, and garages.

Last, we looked at how sensitive subjects were to having different personal aspects observed by surveillance. Questions in the second survey were asked in the first session, before each subject took their walk, regarding:

1. Race

2. Gender

3. Face

The survey had questions about theses identifying features because they are common features mentioned in the surveillance literature for identification and use.

We ran a within-subject, one-way ANOVA to see subjects' comfort level of having these features identified in surveillance varies. The independent variable is the identifying feature, with three levels each representing one of the three specific features listed above. The dependent variable is the comfort level with these features being monitored by surveillance. The results from performing the ANOVA are shown in Figure 9.5.

As can be seen from the results, a particular identifying feature does effect the subject's concern. In particular, the face is the feature most subjects are concerned with being identified under surveillance.

9.2.3 Visual Filters

To discover how subjects feel about visual filtering measures in regard to their privacy expectations, we examine the subject responses on the filters.

First, we looked at how subjects rated their comfort in doing the five activities they performed on the walk by examining the data from the first session survey. This data is a subset of what was already shown in Figure 9.3. As we know, activity does affect comfort level, and the activities we had subjects perform spanned much of the differing levels. Posting flyers was ranked more uncomfortable, taking photos and asking a stranger for information had more average comfort ratings, and eating and reading were comfortable activities. Thus, the activities we had subjects perform on the walk provided a range of comfort levels.

Source	SS	df	MS	F	Prob > F
Columns	94.82	9	10.5356	18.48	0
Error	108.3	190	0.57		
Total	203.12	199			

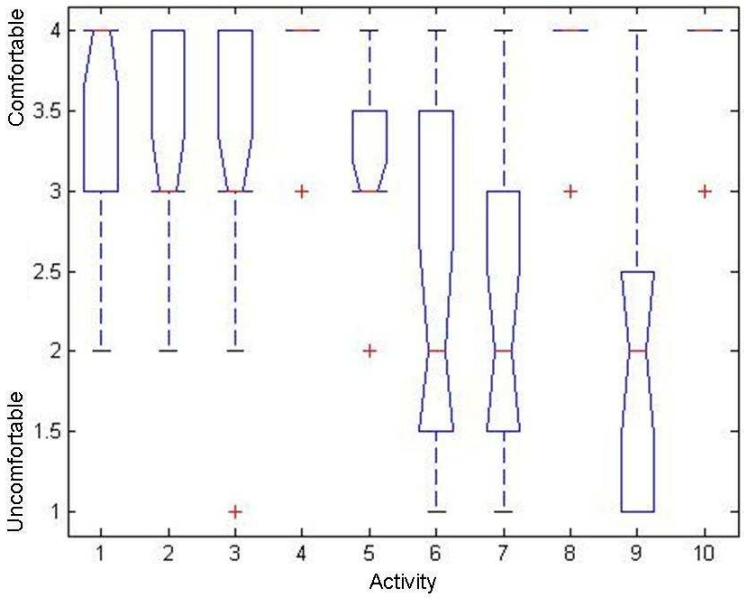


Figure 9.3: Activities: (Top) the ANOVA table for privacy results across the 10 activities (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the activity

Next, we compared activities against the different visual filter measures. For each filter, subjects were asked to rank how comfortable they were performing the five activities with this filter on. We did a within-subject, two-way ANOVA with the independent variables being the activity and filter. For activity, there are five different levels corresponding to the five specific activities. For the filter type, there are four levels corresponding to the four different visual filters. The dependent variable is

Source	SS	df	MS	F	Prob > F
Columns	82.38	9	9.15333	16.66	0
Error	104.4	190	0.54947		
Total	186.78	199			

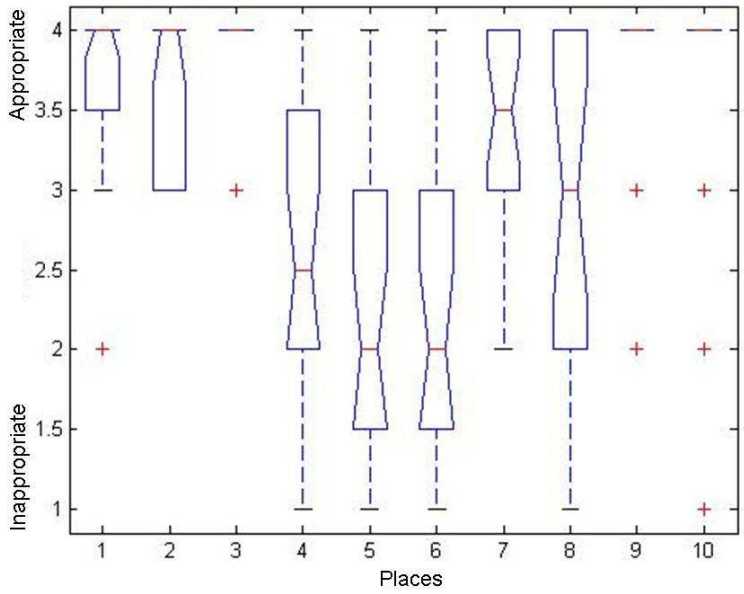


Figure 9.4: Places: (Top) the ANOVA table for privacy results across the 10 places (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the place

comfort. The results from the test are shown in Table 9.1. As it shows, there is no interaction between activity and filter in effecting the comfort. The comfort is affected by filter type alone.

Given these findings, we then performed within-subject, one-way ANOVAS across all the activities, with a given filter on or off. The results the filters are shown Figures

Source	SS	df	MS	F	Prob > F
Columns	16.7937	2	8.39683	9.53	0.0003
Error	52.8571	60	0.88095		
Total	69.6508	62			

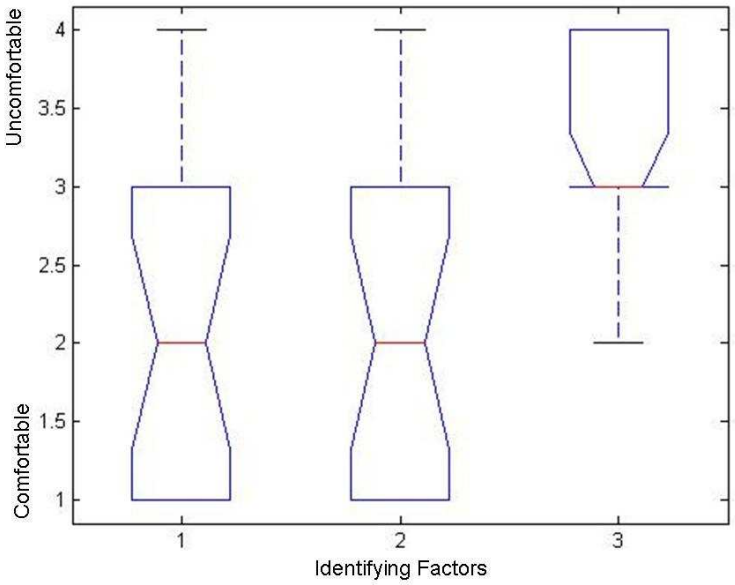


Figure 9.5: Identifying Factors: (Top) the ANOVA table for privacy results across the 3 identifying factors (Bottom) the box plot for the ANOVA results showing that there is a variance in sensitivity depending on the identifying factor

9.6a, 9.6b, 9.6c, and 9.6d.

In looking at the results from these ANOVA tests, it can be seen that the Background, Outline, and Blur filters have an effect on comfort. Having the face filter on or off, however, does not effect comfort.

In order to answer the last research question, we performed within-subject, one-

Source	SS	df	MS	F	Prob > F
Columns	16.883	3	5.63778	19.37	0
Rows	0.681	4	0.17024	0.6729	
Interaction	0.557	12	0.04643	0.16	0.9995
Error	116.19	400	0.29048		
Total	134.312	419			

Table 9.1: Anova Table for Activities

way ANOVA across all the filters types and the subjects' ratings at how well they thought the filters preserved their privacy. The factor is the type of visual filter measure, with four levels, each representing one of the four filter types. The dependent variable is privacy preservation. The results are shown in Figure 9.7a. As can be seen, the type of filter does have an effect on how well subjects feel their privacy is preserved.

Given these effects, we next look at which filter subjects feel is best at preserving their privacy since there is a difference between filters. In the second session, subjects ranked all four filters from most privacy preserving to least privacy preserving. We again performed a within-subject, one-way ANOVA on this rank data with

the independent variable is the type of visual filter measurement. The dependent variable is privacy preservation rank. The results are shown in Figure 9.7b. As can be seen, the background filter receives the best rank for being privacy preserving. Thus, this is the visual filter that subjects feel is most effective.

9.2.4 Discussion

With this study, we set out to evaluate visual privacy in public spaces from the perspective of the observed individual. In particular, we want to know: 1) what are the privacy expectations observed individuals have in public spaces and 2) if the four proposed filtering measures uphold these expectations and how they can be used in embedded design. Here we discuss the analysis of these issues given the results from our data and our research questions.

9.2.5 Privacy Expectations

The first three of our research questions look at the expectations observed individuals have on visual privacy in public spaces.

Question 1: Do common public activities vary in privacy sensitivity?

When subjects were asked to rate ten different common public activities, the results from our survey showed that there is a statistically significant difference between the comfort levels people feel in performing these activities in public. Not all activities that are performed in public are always comfortable for subjects to have others observe. These results tell us that not all public activities should be considered equal and thus, surveiling these activities is not equally acceptable to subjects. Since subjects vary in their sensitivity to having these activities observed in public, having these observed by a wider audience through surveillance is a concern. This gives us an indication that in places where more sensitive activities are likely to occur, it

becomes even more crucial to focus on privacy protection measures.

Even for reading a magazine, an activity subjects felt comfortable doing, there was a statistically meaningful difference between comfort levels reported on the first session for doing this in public and the comfort level they reported on the second session after knowing they had been under surveillance, but before they saw their videos. The rating on comfort once they knew they had been under surveillance was less than the originally reported comfort levels from the first session. While the comfort level of the eating and photo-taking activities showed no statistically significant change, this further indicates that the type of activity does have an effect on comfort at being observed through surveillance. Even some activities that individuals are comfortable performing in public, where people can observe them, are not as comfortable for subjects' to have surveilled.

Question 2: Is there a difference in comfort level with surveillance in different places?

From the survey results, it is clear that in different places individuals have different expectations as to whether they should be surveilled in those places and how. For places considered more enclosed and privately held, such as stores, malls, banks, airports, and parking garages, subjects thought it was appropriate to have surveillance. Thus, there is an expectations on being observed in these areas and it considered acceptable. However, in contrast, the more open, public spaces listed in the survey, such as residential streets, public parks and plazas, and campuses, were ranked by subjects

as inappropriate for surveillance to be present. Subjects do not feel it is appropriate to be watched in these areas by remote surveillance. In these areas then, subjects would be more sensitive to having their actions observed and recorded. There is an expectation that what is done in these public spaces is only observable by those in the vicinity and is not observable through surveillance. This was tested both in the first and second session with no statistical difference between ratings of any of the places and comfort levels between sessions.

Question 3: Which identifying factors are of more concern to individuals being surveilled?

In looking at the results for the identifying factors of race, gender, and face - which are items noted as being observed by camera operators - there was a clear distinction with these factors. Having their gender and race recognized was not nearly as concerning for subjects as having their face identified. One might speculate that the face is a more personal feature, unique for each person, and that this is a more identifying feature than race or gender. Both race and gender ranked at the same level of concern. Thus, in looking at visual privacy, it is more important that the face is unidentifiable in the image as opposed to race or gender.

In looking across all these findings to answer the issue of expectations on privacy that people have in public places, it is clear that there is an expectation on privacy in public spaces regarding surveillance. Even though our subjects are from a generation that grew up immersed in technology, they still are not comfortable with everything

being observed. Activity, place, and identifying features all have an effect and no single aspect can be discounted. Thus, when examining surveillance and the effect a system will have, these aspects must be taken into account. In more open, public spaces, individuals do not think surveillance is as appropriate as in more closed, privately owned spaces. When installing a surveillance system up in a place that is more open and public, such as parks and plazas, what activities and what factors are monitored must be considered. If more personal activities which express opinions and taste, such as political activities and flyer distribution, are going on in a place, these are sensitive activities for observed individuals. As there already is the expectation that surveillance is not appropriate in public spaces, then having these sensitive activities surveilled in a public space is not expected nor deemed appropriate. Additionally, if the surveillance system shows and tracks faces, this is also deemed uncomfortable for subjects and is a privacy concern. Therefore, in designing these systems, trying to account for the privacy expectations of individuals in that particular place, and which factors to focus on, are important.

9.2.6 Filters in Relation to Privacy

In order to try and preserve expectations of visual privacy in public spaces under surveillance, we examined at four different visual filtering measures and how subjects' perceptions of them.

Do subjects feel their privacy is preserved by the four visual filters

presented?

From the results, there was a clear difference between the filtering measures. When looking at filters in conjunction with activity, the comfort level subjects felt depended only on the filter, not the particular activity being performed, even if the activity was deemed as being uncomfortable to perform in public. Thus, the filters can be judged for effectiveness independent of activity. When looking at the results across the filters, it can be seen that the facial obscuring measure provided less comfort as opposed to the other three measures. The outline filter and blur filters were next, with no meaningful statistical significance between them, and the solid silhouette filter was thought to be the most privacy preserving.

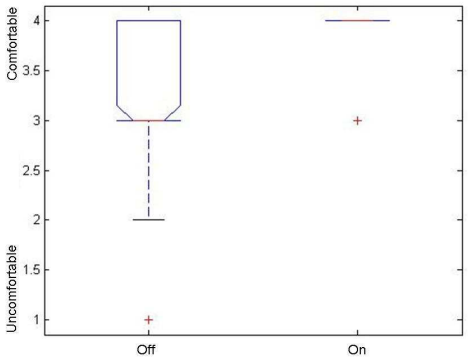
Even though subjects' responses to identifying features indicated that the face was the aspect they wanted the most protection of, from the filter responses we can see that obscuring just the face is not enough. The solid silhouette filter hides not only the face, but other visual details of the observed individual. The solid silhouette filter shows large motions of the observed individual and the shape of the human. The shape of the human can perhaps lead to a determination of gender if more gender identifying features, for example a ponytail on the head, are displayed. However, the race and all small motions or activities are hidden. From the results, this filter provides the level of visual detail that observed individuals are comfortable providing under surveillance in public spaces.

What visual privacy filter of the four do subjects prefer for privacy

preservation?

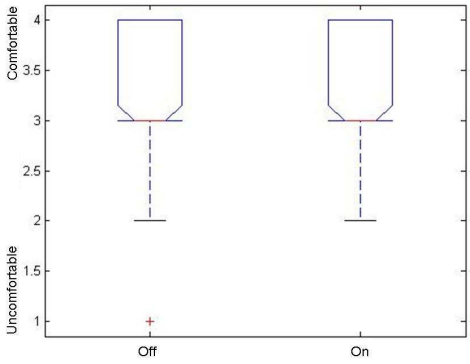
The result from the prior questions showing the solid silhouette being the preferred filter was further supported by the results from the survey when we asked subjects to rank the filters on the order of privacy preservation. The solid silhouette filter was ranked as the most privacy preserving, while the blur filter ranked the next highest, then the outline filter, and then the facial obscuring filter. The facial obscuring filter still provided a detailed level of information about the observed individual other than the face, which is still a concern for visual privacy by subjects. The outline filter still shows a certain level of detail on the face, which is probably why this filter ranks lower than the blur and solid silhouette, even though it hides most of the detailed information on the human, including race, since color is removed.

Source	SS	df	MS	F	Prob > F
Columns	20.743	1	20.7429	46.42	0
Error	92.952	208	0.4469		
Total	113.695	209			



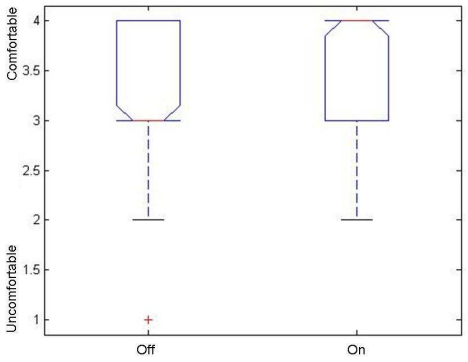
(a) Solid silhouette filter off vs. on

Source	SS	df	MS	F	Prob > F
Columns	0.305	1	0.30476	0.53	0.4692
Error	120.552	208	0.57958		
Total	120.857	209			



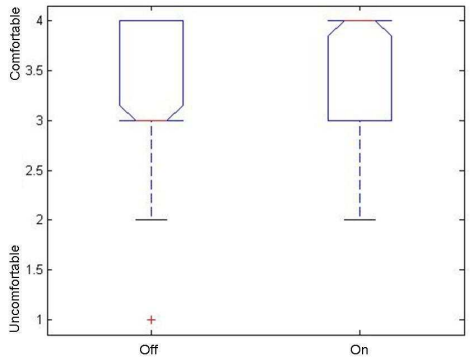
(b) Face obscuring filter off vs. on

Source	SS	df	MS	F	Prob > F
Columns	6.519	1	6.51905	11.05	0.0011
Error	122.762	208	0.5902		
Total	129.281	209			



(c) Comfort with outline filter off vs. on.

Source	SS	df	MS	F	Prob > F
Columns	10.971	1	10.9714	21.25	0
Error	107.41	208	0.5164		
Total	118.381	209			

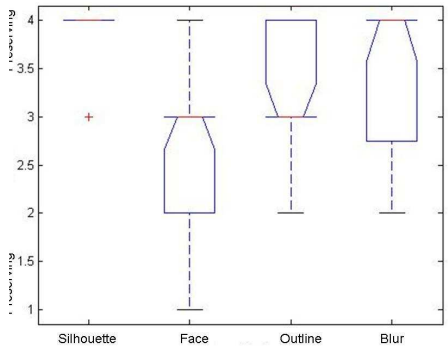


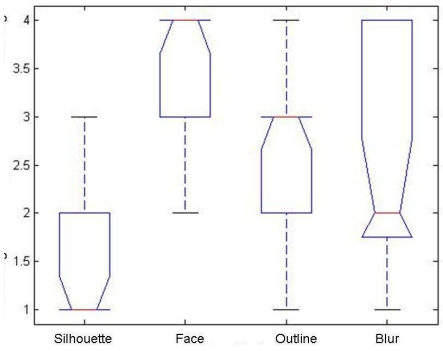
(d) Patched-based blur filter off vs. on.

Figure 9.6: Comfort level with the filter on versus off done for each filter independently. The ANOVA table is shown above the box plot each of the subfigures

Source	SS	df	MS	F	Prob > F
Columns	18.7619	3	6.25397	14.55	0
Error	34.381	80	0.4296		
Total	53.1429	83			

Source	SS	df	MS	F	Prob > F
Columns	50.5238	3	16.8413	24.73	0
Error	54.4762	80	0.681		
Total	105	83			





(a) Independent filter privacy rating **(b)**Overall filter rankings

Figure 9.7: The filter ratings with the ANOVA table shown above the box plot each of the subfigures

Chapter 10

Visual Privacy Discussion

> *Technology is neither good nor bad, nor even neutral. Technology is one part of the complex of relationships that people form with each other and the world around them; it simply cannot be understood outside of that concept.*
>
> Samuel Collins

Visual privacy is a complex subject and our work provides insight into expectations of privacy with public surveillance. While surveillance networks have benefits, we hope to find a way to balance these with privacy expectations. In this final chapter, we discuss where the current technology is for filtering measures and future development directions for these filters. Additionally, we discuss future directions with our work in trying to achieve this balance.

10.0.7 Current State of the Technology and Recommendations

The results of our study suggest that there are expectations of privacy in public spaces and that the visual filtering measures can provide privacy preservation for individuals. Thus, visual filtering measures can be used for embedded design in order to take into account the needs of observed individuals. It is desirable to have embedded privacy preservation measures in place so that observed individuals acting within the space are not inhibited by the camera technology and there remain places where individuals feel comfortable expressing themselves through actions like participating in political activities and posting flyers.

Implementing a filter through embedded design implies that the visual filter must be automatic. The observed individual in the place has no interaction with the system and the camera operators. On the other end of the system, the operator should not have control over when the filter is on or off as they can violate an individual's privacy by turning the filter off. The filter must be run automatically on the images from the cameras without input from the individual or the camera operator. In trying to balance the needs of the camera operators, if it is deemed necessary to view the unfiltered images, then a password or another security mechanism can be used to turn the filter off. However, the default setting should be so the filter is automatically on in order to preserve privacy.

Our findings show that the solid silhouette filter provides individuals with the best

sense of visual privacy preservation. Currently, this sort of filter is not fully automatic. In order to make this type of filter automatic, there is further research that needs to be done in analyzing video. The current state-of-the-art work from the computer vision community provides a few different methods to do this solid silhouette filter, but all these have limitations given certain circumstances. Moving objects in a scene can be segmented from the background using background subtraction methods. A review of background subtraction approaches is given in [71]. These methods work well when objects are in motion, but when the object stops moving, it becomes part of the background and its identity is revealed. It is not suitable to preserve privacy only when individuals are in motion. Thus, background subtraction alone is not enough to do the solid silhouette filter. Other computer vision methods, such as pedestrian tracking and human detection, use shape to find humans, an example of which is discussed in [94, 93]. However, these methods are still not very advanced and typically can only find humans in upright poses, but have difficulty when humans are sitting or are in more articulated poses. Face detection could also be used as a cue to where a human is, but if the human is turned around or if the face is occluded or hard to detect, then face detection will fail. For privacy protection, more research needs to be done in the computer vision community into detecting humans, and possibly how to combine current approaches, in order to make a fully automatic solid silhouette filter.

This gives a direction for vision researchers and industries interested in visual

privacy to focus on. In the mean time, methods such as the blur filter, which was ranked as a second to the solid silhouette filter, can be used automatically. This method is fully automatic at the current time and could be use in embedded design. Based on the environment the cameras are in, the parameters on these algorithms, such as to what level things should be outlined at or what amount of blur should occur, will need to set.

10.1 Towards a Balance with Visual Privacy

As privacy is a very context-sensitive issue, with this work we have looked specifically and privacy in the context of surveillance of public places. We have demonstrated that observed individuals have an expectation of privacy in public spaces which is effected by many differing factors including activity, location, and identifying features. Given this, we have also shown that visual filtering measures are effective for observed individuals at upholding their visual privacy expectations. This work is one step to better understanding visual privacy in relation to camera networks.

It is important to continue analyzing privacy in the setting of public surveillance. In particular, looking at how we can balance the needs of the camera operators and observed individuals with different measures, such as technology. While policies by governing bodies can help set the regulations for camera networks, technical innovations can help uphold such regulations while balancing the privacy needs of the observed individuals.

In future directions of this work, we will explore the needs of camera operators and how they are using their surveillance cameras. We will focus on police use of surveillance and how they interpret the image data for their use. We will then see how we can balance their needs with privacy expectations in public settings and whether technology measures, such as visual filters, can be beneficial doing so.

391

Chapter 11

Conclusion

This dissertation started by looking at the way camera networks have changed over time and the current state of camera networks which have become more pervasive, larger, and able to capture and store dynamic data more easily. These changes have led to challenges in data correlation, data integrity, and data privacy. The substantive sets of chapters dealt with a specific project involving each of these various notions, respectively, given the current state of camera networks: localization using dynamic scenes, attack detection on cameras in uncontrolled locations, and a study of visual privacy expectations in public settings with surveillance. With this work we have provided some steps to improve the way current camera networks operate and using dynamic scene data for applications.

We hope that our work will inspire others to continue examining the area of camera networks, and multi-view settings, and go beyond the single image to look at

the dynamic information that is becoming so prevalent. Video data provides a rich source of information that is only starting to be explored. Additionally, we hope to inspire others to continue working in the intersection of technology and society and be part of the process to uphold the balance between man and machine.

Bibliography

[1] City of Sydney Street Safety Camera Program: Code of Practice.

[2] EasyCal Camera Calibration ToolBox. http://www.cis.upenn.edu/teleimmersion/research/downloads/EasyCal/.

[3] Television on a disk. *TIME*, September 1972.

[4] New orleans mayor launches technology recovery. City Of New Orleans Mayors Office of Communications Press Release, Nov. 29 2005.

[5] Q & A: the congestion charge. *The Guardian*, 2006.

[6] ACLU-NC. New technology blurs surveillance and privacy.

[7] A. Aksay, A. Temizel, and A.E. Cetin. Camera tamper detection using wavelet analysis for video surveillance. *IEEE Conference on Advanced Video and Sginal Based Surveillance*, 2007.

[8] J.P Anderson. Computer security threat monitoring and surveillance. *Technical report, James P Anderson Co., Fort Washington, Pennsylvania*, April 1980.

[9] Monroe Anderson. Picture this: Aldermen caught on camera. *Chicago Sun-Times*, Jan. ,14 2006.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(4):509–522, April 2002.

[11] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. *Proceedings of Workshop on Motion and Video Computing*, 2002.

[12] M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. *In the Proceedings of the Conference on Computer Supported Cooperative Work*, 2000.

[13] S. Buchegger and J. Le Boudec. Performance analysi of the CONFIDANT protocol: Cooperation of nodes-fairness in dynamic ad-hoc network. *IEEE/ACM symposium on mobile Ad Hod Networking and Computing*, 2002.

[14] O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *In BMVC*, 2002.

[15] City of Sydney Australia Government. Cctv.

[16] J.B. Collins and J.K. Uhlmann. Efficient gating in data association with multivariate distributed states. *IEEE Trans. Aerospace and Electronic Systems*, 28(3):909–916, July 1992.

[17] J. Connell, A.W. Senior, A. Hampapur, Y-L Tian, L. Brown, and S. Pankanti. Detection and tracking in the IBM peoplevision system. *IEEE ICME*, 2004.

[18] I.J. Cox and S.L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In *International Conf. on Pattern Recognition*, pages 437–443, 1994.

[19] R. Cucchiara, M. Piccardi, and P. Mello. Image analysis and rule-based reasoning for a traffic monitoring system. *EEE Trans. Intelligent Transportation Systems*, 1(2):119–130, June 2000.

[20] R. Cucchiara, A. Prati, and R. Vezzani. A system for automatic face obscuration for privacy purposes. *in Pattern Recognition Letters*, 2006.

[21] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[22] Walter Dornberger. *V-2.* The Viking Press, 1954.

[23] M. Gill et. al. The impact of CCTV: Fourteen case studies. *Home Office of the United Kingdom Government*, 2005.

[24] M. Fan, Y. Tan, and A.B. Whinston. Evaluation and design ofonline cooperative feedback mechanisms for reputationmanagement. *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[25] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, Sept. 1991.

[26] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar. Distributed localization of networked cameras. *The Fifth International Conference on Information Processing in Sensor Networks*, April 2006.

[27] By Ray Furlong. Germans probe merkel spy camera. *BBC News, Berlin*, March 27 2006.

[28] S. Ganeriwal and M.B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Security for Ad-hoc and Sensor Networks*, 2004.

[29] Charles P. Ginsburg. *The birth of video recording.* Ampex Corporation, 1980.

[30] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. *Proc. Fourth European Conf. Computer Vision*, 1996.

[31] Marianne L. Gras. The legal regulation of CCTV in europe. *Surveillance & Society*, 2004.

[32] Ann Harrison. Hackers rebel against spy cams. *WIRED*, Dec. 29, 2005.

[33] Information Commissioner. CCTV code of practice, July 2000.

[34] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *International Conference on Computer Vision*, pages 34–41, 2001.

[35] A. Josang and R. Ismail. the beta reputation system. *Bled Electronic Commerce Conference*, 2002.

[36] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Trans. Intelligent Transportation Systems*, 1(2):108–118, June 2000.

[37] Jay W. Summet Khai N. Truong, Shwetak N. Patel and Gregory D. Abowd. Preventing camera recording by designing a capture-resistant environment. *Ubicomp*, 2005.

[38] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, Nov. 2005.

[39] I. Kitaharaa, K. Kogure, and N Hagita. Stealth vision for protecting privacy. *International Conference on Pattern Recognition*, 2004.

[40] T. Koshimizu, T. Toriyama, and N. Babguchi. Factors on the sense of privacy in video surveillance. *The 3rd ACM Workshop on Capture, Archival and Retrieval of Personal Experiences (CARPE)*, 2006.

[41] Branislav Kusy, Akos Ledeczi, Miklos Maroti, and Lambert Meertens. Node density independent localization. In *International Conference on Information Processing in Sensor Networks (IPSN 2006)*. ACM, 2006.

[42] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Networks*, 43(4):499–518, 2003.

[43] L. Lee, Raquel Romano, and Gideon Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[44] Wei Li, Zhengquan Xu, and Ye Yao. An efficient scheme to secure vlc codeword concatenations for video encryption. volume 5960, page 59602T. SPIE, 2005.

[45] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 2004.

[46] Y. Ma, S. Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3D Vision*. Spinger-Verlag, 2004.

[47] W.E. Mantzel, Choi Hyeokho, and R.G. Baraniuk. Distributed camera network localization. *Asilomar Conference on Signals, Systems and Computers*, Nov. 2004.

[48] Miklós Maróti, Branislav Kusý, György Balogh, Péter Völgyesi, András Nádas, Karoly Molnár, Sebestyén Dóra, and Ákos Lédeczi. Radio interferometric geolocation. November 2005.

[49] O. Masoud, N. P. Papanikolopoulos, and E. Kwon. The use of computer vision in

monitoring weaving sections. *IEEE Trans. Intelligent Transportation Systems*, 2(1):18–25, March 2001.

[50] M. McCahill and C. Norris. From cameras to control rooms: the mediation of the image by cctv operatives. *CCTV and Social Control:The politics and practice of video surviellance-European and global perspectives*, 2004.

[51] Michael McCahill and Clive Norris. Cctv in london. *Urban Eye Project*, 2002.

[52] M. Meingast, M. Kushwaha, X. Koutsoukos S. Oh, A. Ledeczi, and S. Sastry. Heterogeneous camera network localization using data fusion. *ACM/IEEE International Conference on Distributed Smart Camera*, 2008.

[53] M. Meingast, S. Oh, and S. Sastry. Automatic camera network localization using object image tracks. *IEEE 11th International Conference on Computer Vision*, 2007.

[54] P. Michiardi and R. Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad-hoc networks. *Conference on Communication and Multimedia Security*, 2002.

[55] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International journal of Computer Vision*, 2004.

[56] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[57] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *In the International Journal of Computer Vision*, 65:43–72, 2005.

[58] Martha T. Moore. Cities opening more video surveillance eyes. *USA Today*, July 18, 2005.

[59] J. Mundinger and J. Le Boudec. Analysis of a robust reputation system for self-organized networks. *University of Cambridge*, 2004, January Statistical Laboratory Research Report.

[60] C. Neustaedter and S. Greenberg adn M. Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Transactions on Computer-Human Interactions*, 2006.

[61] E. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *Transactions on Knowledge and Data Engineering*, 2005.

[62] NYCLU. NYCLU report documents rapid proliferation of video surveillance cameras, calls for public oversight to prevent abuses, December 13, 2006.

[63] Office of the Privacy Commissioner of Canada. OPC guidelines for the use of video surveillance of public places by police and law enforcement authorities.

[64] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain Monte Carlo data association for general multiple-target tracking problems. In *Proc. of the*

43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, Dec. 2004.

[65] Songhwai Oh, Stuart Russell, and Shankar Sastry. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Trans. Automatic Control (submitted)*, 2007.

[66] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, 2004.

[67] J.-M. Chassery P. Bas and B. Macq. Robust watermarking based on the warping of predefined triangular patterns. *Security and Watermarking of Multimedia Contents II*, 2002.

[68] E. Friedman P. Resnick, R. Zeckhauser and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[69] Eric Pape. More watchful eyes on the continent. *Newsweek:International Edition*, Feb. 20 2006.

[70] F. A. P. Petitcolas. Watermarking schemes evaluation. *IEEE Signal Processing*, 17:58–64, Sept. 2000.

[71] Massimo Piccardi. Background subtraction techniques: a review. *IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[72] A.B. Poore. Multidimensional assignment and multitarget tracking. In Ingemar J. Cox, Pierre Hansen, and Bela Julesz, editors, *Partitioning Data Sets*, pages 169–196. American Mathematical Society, 1995.

[73] M. Jakubowski R. Venkatesan, S. M. Koon and P. Moulin. Robust image hashing. *EEE International Conference on Image Processing*, 2000.

[74] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. *Computer Vision and Pattern Recognition*, 1, July 2004.

[75] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24(6):843–854, December 1979.

[76] P. Resnick and R. Zeckhauser. Trust among strangers ininternet transactions: Empirical anlysis of ebay's reputationsystem. *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce*, November 2002.

[77] R.Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[78] E. Ribnick, S. Atev, O. Masoud, N. Papanikolopoulous, and R. Voyles. Real-time detection of camera tampering. *IEEE International Conference on Video and Sginal Based Surveillance*, 2006.

[79] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Goldberg. Respectful

cameras: Detecting visual markers in real-time to address privacy concerns. *International Conference on Intelligent Robots and Systems*, 2007.

[80] Changgui Shi and Bharat Bhargava. A fast mpeg video encryption algorithm. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 81–88, New York, NY, USA, 1998. ACM Press.

[81] R.W. Sittler. An optimal data association problem on surveillance theory. *IEEE Trans. Military Electronics*, MIL-8:125–139, April 1964.

[82] Paul Stanton, William Yurcik, and Larry Brumbaugh. Protecting multimedia data in storage: a survey of techniques emphasizing encryption. volume 5682, pages 18–29. SPIE, 2005.

[83] D.E. Stevenson and M.M Fleck. Robot aerobics: four easy steps to a more flexible calibration. In *International Conference on Computer Vision*, Paradise Island, Bahamas, June 1995.

[84] Bob Sullivan. Privacy under attack, but does anybody care? *MSNBC*, Oct. 17 2006.

[85] Kevin Sullivan and Karla Adam. Jury sees tape from london bomb scare. *Washington Post*, Jan. 16 2007.

[86] A. Zisserman T. Kadir and M. Brady. An affine invariant salient region detector. *In ECCV*, 2004.

[87] A. Zisserman T. Kadir and M. Brady. An affine invariant salient region detector. *In ECCV*, 2004.

[88] J.C. Tai, S.T. Tseng, C.P. Lin, and K.T. Song. Real-time image tracking for automatic traffic monitoring and enforcement applications. *Journal of Image and Vision Computing*, 22(6):485–501, June 2004.

[89] The Constitution Project. Guidelines for public video surveillance: A guide to protecting communities and preserving civil liberties. Technical report, The Constitution Project's Liberty and Security Initiative, 2006.

[90] T.Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 2004.

[91] Nicholas J. Wade and Stanley Finger. The eye as an optical instrument: from camera obscura to helmholtz's perspective. *Perception*, 20(10):1157–1177, 2001.

[92] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian. Privacy protecting data collection in media spaces. *ACM Multimedia (MM)*, 2004.

[93] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *Transactions on PAMI*, 19(7):780–785, 1997.

[94] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *International Conference On Computer Vision*, 2005.

[95] C. W. Wu. On the design of content-based multimedia authentication systmes. *IEEE Transactions on Multimedia*, 2002.

[96] J. Kosecka Y.Ma, S. Soatto and S. Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, 2004.

[97] Y.Tian, M. Lu, and A. Hampapur. Robust and efficient foreground analysis for real-time video surveillance. *IEEE Conference on Vision and Pattern Recognition*, 2005.

[98] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[99] Z.Zivkovic. Improved adaptive gausian mixture model for background subtraction. *International Conference Pattern Recognition*, 2:28–31, 2004.

[100] Z.Zivkovic and F.van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.