A FRAMEWORK FOR ANONYMOUS SURVEILLANCE

BAKUL TREHAN

UNIVERSITY OF WINNIPEG

2015

A FRAMEWORK FOR ANONYMOUS SURVEILLANCE

By

BAKUL TREHAN

A thesis submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements for the
Master of Science degree.

Department of Applied Computer Science

Master of Science in Applied Computer Science and Society Program

The University of Winnipeg
Winnipeg, Manitoba, Canada
December 2015

A FRAMEWORK FOR ANONYMOUS SURVEILLANCE

BAKUL TREHAN

2015

Thesis Committee

- Dr. Pradeep K. Atrey (Thesis advisor)

  (*On leave from*) Department of Applied Computer Science

  The University of Winnipeg, Winnipeg, MB, Canada

  (*Presently at*) Department of Computer Science

  State University of New York, Albany, NY, USA

- Dr. Shakhawat Hossain (Internal examiner)

  Department of Mathematics and Statistics and

  Department of Applied Computer Science

  The University of Winnipeg, Winnipeg, MB, Canada

- Dr. Yang Wang (External examiner)

  Department of Computer Science

  University of Manitoba, Winnipeg, MB, Canada

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Pradeep K. Atrey for his guidance, encouragement and gracious support throughout the course of my work. His expertise motivated me to write this thesis. His mentor-ship was paramount in providing me with in-depth knowledge of the chosen research topic. His constructive criticism and direction has added value to my research. I would also like to thank Dr. Mukesh K. Saini for his support and guidance throughout my research. His suggestions helped me a lot in the development of this system. I feel incredibly lucky to have worked under their supervision.

A special thanks to Dr. Sergio Camorlinga and Dr. Yangjun Chen, who provided me an opportunity to take their graduate courses and inculcated an interest for algorithms and Biometrics. I would like to thank the Department of Applied Computer Science at University of Winnipeg, especially Dr. Simon Liao, Chairman of the Department. I would also like to thank Connie Arnhold, Deanna England, Eric Benson, James Deng, and Dagmawit Habtemariam for their help related to administrative, technical and monetary queries.

I am indebted to the University of Winnipeg in making my dream come true for pursuing higher education in Canada, and for providing The Univer-

sity of Winnipeg Graduate Scholarship through Faculty of Graduate Studies.

Completing this work would have been all the more difficult were it not for the support and friendship of the cheerful group of colleagues at the Graduate Study Lab. One could not have asked for better housemates than Ashmeet and Gagan, who have been very helpful throughout this two year journey. I have gained in a way through their constant geek talks about vectors and algorithms. Shy (Qianjia Huang) has been very motivating by showing strong dedication towards the work and how can I forget about the ever awesome hotpots. I have made some lifelong friends like Abukari Yakubu, Manish Sharma, Orlando Simpson, Harmeet Singh, Kanwarpreet Kaur, Aditya Bhardwaj, and Ashish Tripathi.

I would sincerely like to thank my family as they stood with me through thick and thin. I really thank them for keeping faith in me. I would like to thank the many participants (and their families) who took part in the studies that make up this thesis. They volunteered a great deal of time, energy and without them none of these studies would have been possible.

Last, but by no means least, I thank *Lord Hanuman* for his blessings, for he is the ultimate reason behind all happiness and content in my life. For all those I might have not mentioned, consider my explicit thanks!

Contents

Abstract

Preserving the privacy of people in video surveillance systems is quite challenging and a significant amount of research has been done to solve this problem in recent times. Majority of existing techniques are based on detecting bodily cues such as face and/or silhouette and obscuring them so that people in the videos cannot be identified. We observe that merely hiding bodily cues is not enough for protecting identities of the individuals in the videos. An adversary, who has prior contextual knowledge about the surveilled area, can identify people in the video by exploiting the implicit inference channels such as behavior, place and time. This thesis presents an anonymous surveillance framework which advocates for outsourcing of surveillance video monitoring (similar to call centers) to the remote sites where professional security operators watch the video and alert the local site when any suspicious or abnormal event takes place. We find that remote monitoring helps decoupling the contextual knowledge of security operators. Since security operators at the remote site could turn into adversaries, a trust computation model to determine the credibility of the operators is presented as an integral part of the proposed framework. In order to test the feasibility of the framework, this thesis also presents an implementation of a remote surveillance system. The experiments suggest that the proposed framework/system

provides more robust measures of privacy yet maintaining the surveillance effectiveness.

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Extension
APPEL	A P3P Preference Exchange Language
CCTV	Closed-Circuit Television
CERN	European Organization for Nuclear Research
DARPA	Defense Advanced Research Projects Agency
DCT	Discrete Cosine Transform
FLV	Flash Video
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IIS	Internet Information Services
JPEG	Joint Photographic Experts Group
JPEG XR	JPEG Extended Range
MPEG	The Moving Picture Experts Group
.net	Dot Net
PASS	Privacy Aware Surveillance System
PET	Privacy Enabling Techniques
P3P	Privacy Preference Project
RC4	Rivest Cipher 4

RFID	Radio Frequency Identification
RTMP	Real Time Messaging Protocol
RTMPE	Enhanced Real Time Messaging Protocol
RTMPS	Secure Real Time Messaging Protocol
RTMPT	Real Time Messaging Protocol Tunneled
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security/Secure Sockets Layer
WWW	World Wide Web
2D	Two Dimensional
3D	Three Dimensional

List of Symbols

Symbol	Definition
A	Set of contextual attributes
a_i	An attribute in set A
b	Weight given to the reward/punishment
$\mathcal{C}_{opl}$	Contextual overlap
g	Number of incorrectly identified events
$\mathcal{K}^h$	Binary context vector from operator's prior knowledge
$\mathcal{K}_{max}$	Maximum value of probability of privacy loss
$\mathcal{K}_{opl}$	Probability of privacy loss
$\mathcal{K}^q$	Binary context vector based on questionnaire
m	Max value of x (equals 5)
n	Number of correctly identified events
o	Operator
$P_{(t)}$	Current value of punishment
$R_{(t)}$	Current value of reward
t	Current time
x	Extension number (can be 1 to 5)
α_p	Parameter to control punishment (equals 1)

α_r	Parameter to control reward (equals 0.5)
ϵ	Very small value which accounts for zero C_{opl}
μ	A coefficient that depends on the unit chosen for time
$\phi_{(t)}$	Current trust level of an operator
$\phi_{(t-w)}$	Previous trust value of an operator
$\odot$	Element-wise OR operation

Chapter 1

Introduction

Video surveillance has gained a lot of popularity over the past few years. We can find surveillance systems in almost all public places, such as airports, schools, universities, banks and shopping malls. They are used to record and monitor the activities of people. Surveillance is required for public safety as it helps us in preventing crimes. However, it also has some drawbacks associated to people's privacy.

While various privacy protection methods have been introduced in the past to protect the privacy of individuals present in the surveillance videos, privacy loss remains a serious issue associated with surveillance. In the past, researchers have developed methods for hiding the facial features to protect the privacy of individuals, but hiding facial features can render the surveillance video useless. We need a robust privacy protection system which allows us to perform the surveillance tasks effectively as well as protect the privacy of individuals.

In the rest of this chapter, we review the current surveillance systems

and discuss what motivated us to pursue the work presented in this thesis in Section 1.1. Next, in Section 1.2, we discuss the drawbacks of current surveillance systems and present our observations regarding the privacy loss in these systems. Further, in Section 1.3, we highlight the goals and contributions of this thesis. Finally, Section 1.4 describes the organization of the thesis.

1.1 Motivation

Current privacy protection methods for surveillance applications usually propose to obscure bodily cues to hide the identity of individuals appearing in the videos. However, Saini et al. [1] highlighted that hiding appearance based cues is not sufficient for protecting one's identity. Other regions of the video, which provide *what* (activity), *when* (time) and *where* (location) information can also cause identity leakage of the individuals.

Generally, we find that it is extremely difficult to hide identities in traditional video surveillance systems. The security operators (or simply operators) who monitor the camera views generally not only have a good knowledge of the surveillance site and current context, they are familiar with the people who regularly appear in the videos and are therefore more likely to be able to identify them. This leads to privacy loss. The main motivation of the proposed work is to address the issues discussed above that occur in traditional surveillance systems and to provide a more robust, privacy-protected surveillance solution.

1.2 Privacy Challenges in Current Video Surveillance Systems

In traditional video surveillance systems, cameras are placed in important locations and video feeds are sent to local security offices. Because automated surveillance is still in its infancy, most of the time these feeds are manually monitored by the security operators. This can cause privacy loss if the security operator (or any other person who has access to the video feeds) acts as an adversary and attempts to identify the individuals in the video, and acquire additional knowledge about them, e.g., habits, company, and other lawful activities.

Privacy is a subjective matter, which means that in some cases an individual may feel that privacy loss has occurred when his or her face is captured by a surveillance camera. On the other hand, some people would feel privacy loss has only occurred when they are recorded by a surveillance camera leading to leakage of any *sensitive* information. For instance, they may not feel any privacy loss has occurred when they are recorded in a public place such as a shopping mall or an airport, but they would mind being recorded in a hospital because that may leak their health-related information. The privacy loss computation model proposed by Saini et al. in their recent work [2] captures these aspects.

The following are our observations regarding the privacy loss in traditional surveillance systems, in which an operator manually watches the camera views.

Observation # 1: Sensitive information cannot be removed

Privacy loss usually occurs when the identity of a person is associated with his/her sensitive information. The first approach for providing privacy protection could be to remove sensitive information from the video. There are two problems with this approach which make it infeasible: (1) it is generally difficult to automatically detect the sensitive information, and (2) the sensitive information is generally important for surveillance purposes, and removing all sensitive information may render the data useless. The second problem arises because both sensitive incidents and suspicious incidents have the common characteristics of high entropy, which are difficult to distinguish. An incident removed from the video as a sensitive incident has a high probability of containing suspicious information. For example, a person's privacy may be sensitive to the objects s/he carries, but these objects may be weapons. Therefore, identity blocking is a more feasible solution of privacy preservation in surveillance systems.

Observation # 2: The *who* information can be hidden through computer vision

Saini et al. [1] proposed that identity leakage occurs through the disclosure of four types of information: *who*, *what*, *when*, and *where*, which are called evidences. While the *who* evidence uniquely identifies a person, e.g., facial information, the *what*, *when*, and *where* information can be used to associate a person with a specific group of people, depending on the evidences obtained and on the adversarys knowledge. In this work, we assume that the who evi-

dence can be removed using computer vision techniques, or in the worst case, the whole image can be globally transformed to hide the facial information. Even if the face hiding technique is not 100% accurate, the presence of faces in the video does not cause privacy loss if the adversary is not able to identify them. In the proposed work, the probability of face identification is further minimized through context decoupling.

Observation # 3: The *what* information is more important for surveillance and does not cause significant privacy loss when detected in isolation

Hiding the *what*, *when*, and *where* information from the local security operator is very difficult, if not impossible. Furthermore, it is not practical to remove the *what* information from the video, because this will defeat the basic purpose of surveillance. One needs to detect the suspicious events and activities in order to assess the security threats, which requires the *what* information to be present in the video. However, the majority of surveillance tasks do not require a person's identity to be known e.g. intrusion, fight, quarrel, and theft. A list of important security threats and corresponding surveillance tasks is provided in Table 1.1. Unlike the *what* information, the *where* and *when* information is less crucial for surveillance; and it can be easily anonymized by providing rules like "At this place people are not allowed to do activity x" or "In this camera view people are not supposed to stand for more than 5 seconds". Fortunately, the *what* information alone does not cause any significant privacy loss [1] if the *when*, and *where* information is not present. Therefore, we turn our focus to block the *when* and

Table 1.1: The surveillance task and associated security threats.

Task	**Respective security threat**
Change detection	Vandalism, Camera tampering, Graffiti
Object introduced	Abandoned baggage, Illegal parking
Object removed	Theft, Museum surveillance
Direction of motion	Counter flow, One way, Immigration
Movement from A to B	Intrusion, Illegal turns, Walking patterns
Cross a zone multiple times	Car surfing, Loitering, Counting
Loitering in a zone	Loitering in a crowd
Overcrowding	Train platforms, Ticket halls
Congestion	Traffic
Sound	Aggression, Fight, Assault
Sound	Gun shots, Firearms
Sound	Panic scream, Shouts, Cry for help
Sound	Vehicle, Lorry, Tank, Airplane
Counting	Vehicles and people venue occupancy, Flow rates, Queue management
Boundary	Perimeter breaches, fence surveillance
Target count	Tailgating, Queue length
Quick movements	Quarrel, Fight
Object association	Gun, Knife, Other weapons
Quick crowd movements	Stampede, Emergency evacuation
Smoke	Fire, Illegal smoking

where information.

Observation # 4: The *where* and *when* information is generally available to the security operator as prior knowledge or through the video content

As mentioned earlier, video feeds are generally monitored locally and the security operator is provided with the location information of the video. Determining the current time is a trivial task as the operator and the camera are in the same time zone. The operator can associate this *when* and *where*

information with the *what* information in the video and infer the identities of the people in the video, even when bodily cues are hidden. In order to protect this identity leakage, the operators should not be provided with the *when* and *where* information, which is very difficult to omit in traditional surveillance systems.

Furthermore, the operators can easily learn the *when* and *where* information from the video even when they are not provided explicitly. It is very difficult to obfuscate the video to hide the location information because the security operators are generally very familiar with the surveillance location. Hiding text legends, symbols, or logos is easier but it is not effective in the case of traditional surveillance systems. To effectively hide the location information from the local operator, a large portion of the image needs to be hidden which is not suitable for surveillance applications. In the proposed work, we found that if the video is monitored remotely by a security operator who has minimal contextual overlap with the surveillance site, it is easy to hide the location information.

1.3 Thesis Goals and Contributions

Our goal was to develop a system that takes into consideration all the above mentioned drawbacks and observations. The work proposed in this thesis is based on the results from Saini et al.'s recent work [3], in which a user study was performed with 10 local users and 7 remote users. They were asked to play the role of a malicious operator whose job was to watch the surveillance videos and identify the people, places and times in them. It was found that

most local users were able to identify people, locations and times whereas remote users found it quite difficult. This leads to the privacy loss of the people appearing in the videos. It has also been found that people feel that their privacy is compromised more when they are monitored locally than when they are monitored from a remote place [4]. Furthermore, the study also revealed that it is very difficult to hide *when* and *where* information from a local operator, therefore the implicit channels (*what*, *when*, and *where*) can cause significant privacy loss [1].

This thesis presents an anonymous surveillance framework, which proposes showing the videos to a person who is situated at a distance sufficiently far away and who does not have contextual knowledge about the surveillance site. Before sending videos to the remote location, the data is transformed in such a way that location and time information cannot be learned from the video. Even though these operators are located at a sufficient distance from the local site, some of them can take interest in knowing more about the site or the people appearing in the videos. They can act as adversaries and can cause privacy loss. We need a *trust model* to "prevent privacy loss", "decide how credible these remote operators are" and "further strengthen our system". While in [3], the idea of anonymous surveillance was proposed and it was substantiated with a user study based experiment, this paper specifically focuses on the design issues of the framework such as "context decoupling", "appropriate video switching time" and "operator trust model". We also present extensive results and analysis based on the experiments.

The main contribution of this thesis is a novel anonymous surveillance framework that advocates for distributed remote monitoring of the video to

preserve the privacy of people. The proposed framework:

- includes a model to compute the contextual overlap, which helps in decoupling the prior knowledge of the operator from the video;
- adopts a time-based random camera assignment strategy to decouple the contextual knowledge of the operators; and
- implements a trust model to determine the credibility of the operators.

1.4 Organization of the Thesis

The work in this dissertation is subdivided into the following chapters: We discuss previous work done on privacy protection as well as observations made based on the previously proposed privacy models in Chapter 2. An overview of the proposed framework and its components such as context decoupling and operator trust model are discussed in Chapter 3. In Chapter 4 we discuss the prototype implementation of the proposed framework, describe the dataset used and present the system architecture along with the data flow diagrams. A discussion of all the experiments performed as well as extensive results based on these experiments are presented in Chapter 5. We conclude the thesis in Chapter 6 and discuss future extensions of the work. We provide further details on the dataset, protocols, video formats and libraries that we used for our proposed work in Appendix A.

Chapter 2

Background and Literature Review

Privacy preserving surveillance has been an active research area for many years. There have been many efforts to hide the identities of the individuals being monitored. Most of these works are based on computer vision techniques to detect and hide bodily cues of identity inference. This chapter brings forward the past works on privacy preserving approaches in video surveillance and identifies the research gaps that are the basis of the work proposed in this thesis. In Section 2.1 we discuss the past works that implemented privacy preserving techniques to protect the privacy of individuals in video surveillance scenarios and highlight how our work is novel compared to existing works. In Section 2.2, we present the chapter summary.

2.1 Previous Work

Hudson and Smith [5] based their work on the dual tradeoff between privacy and awareness, and between awareness and disturbance. In more basic terms, they stated that the more information you receive about someone, the more aware you become of their activities. However, if you keep receiving this information, it can become a disturbance to your normal work. They also devised four techniques: the shadow view technique, a shared audio technique, the synthetic group photo and the "when did Keith leave" technique to solve this issue of dual tradeoff. Their work was based on modifying the whole image rather than just obscuring the faces in the images.

Similar to [5], Boyle et al. [6] also worked on transforming the whole video. They analyzed how a blur and pixelize video filter might impact both awareness and privacy in a media space. They applied a total of 9 filter levels, ranging from heavily applied filters that mask all information, to lightly applied filters that reveal everything. They also designed 3 questionnaires to examine how many awareness cues can be extracted from the filtered video scenes. Some of the awareness cues that they used were: the number of actors; their posture (moving, standing, seated); their gender; the visible objects (basic to detailed), and how available people look (their busyness, seriousness and approachability). They also examined the privacy preserving potential of each filter level. They did a user study with 5 different video scenes. Finally, they concluded that the blur filter at level 5 combined with the pixelize filter around level 6 is a good privacy preserving filter.

A novel privacy aware surveillance system (PASS) was proposed by Barhm

et al. [7]. They worked on completely hiding the facial features and bodily cues. A previously proposed P3P-APPEL framework for managing data privacy on the web was used by the system to encode privacy preferences. They also extended the P3P-APPEL method to make it suitable for video surveillance applications. In the PASS, the users were able to interact with the system with basic hand gestures to gain access to one of the three privacy settings: L0 (no privacy), L1 (face blur), L2 (full body blur). They used Haar-object classifiers for face detection. They also used Eigen faces for face recognition. Once the face is recognized, the user can ask for any level of privacy. They used P3P-APPEL so that the user can communicate with the system after the face recognition. Only those who are recognized have access to privacy levels; others are treated as being unknown and cannot access the privacy levels.

Similar to [7], Wickramasuriya et al. [8] proposed another privacy preserving surveillance technique which used RFID to decide, the privacy level assigned to the people in a video scene. RFID's were used as identity cards. If a user has an RFID with a specific privacy level, it is displayed as a bounding box on the video monitor screen; otherwise, a normal video was displayed. This work was different from previous works because they used RFID's to identify the people and to hide them completely. Instead of using blur or pixelization techniques, they chose to hide the identity by using a bounding box in place of registered people.

Zhang et al. [9] proposed a framework for storing privacy information in a surveillance video as a watermark. In this work, the authorized persons were not only removed from the video but, were also embedded into the video

itself. The original video was only recreated with the help of a secret key. They used a DCT-based perceptual model to create two high capacity video watermarking algorithms. The main goal of this work was to protect the privacy of authorized individuals.

Cheung et al. [10] combined the ideas of Wickramasuriya et al. [8], and Zhang et al. [9], and proposed a privacy protected surveillance system. The system had 3 prime components. First, the selected individuals were identified with the RFID system. This RFID system relays the information to multiple cameras, which in turn track, segment, identify and remove the visual objects corresponding to the individuals with RFID tags. An object-based video impainting algorithm was used to fill in the empty region, which then creates the protected video. Original visual objects are then encrypted and embedded into the compressed protected video. To achieve this, they used a rate distortion data hiding algorithm. They also implemented a data management system which users can use to grant access to the private information to the authenticated clients in a secure and anonymous setting.

Similar to other works, Kitahara et al. [11] also worked on hiding facial features and implemented an anonymous video capturing system, called "Stealth Vision" that protects the privacy of objects by fading out their appearance. They used 3D and 2D positioning of objects to calculate the fade out regions. 3D regions were selected from an overhead camera, whereas the 2D position was captured using cell phone cameras. The position of the objects in the 3D plane was then estimated and it was projected onto a 2D space to get the exact location of the fading out areas.

Chen et al. [12] proposed a system to protect the privacy of specific in-

dividuals in a video recording by hiding facial and bodily features. They address the following two problems: automatic identification of people with limited labeled data, and human body obscuring with preserved structure and motion information. To address the first problem, they proposed a discriminative learning algorithm. This algorithm used labeled training data from the original video and imperfect pair wise constraints labeled from face obscured data to improve the identification accuracy. For the second problem, they proposed a body obscuring method which preserves rich structure and motion information while removing the appearance information of people.

Unlike others, Chaudhari et al. [13] described a real-time privacy protection technique in a life-log video and worked on hiding the facial features as well as distorting the speech in videos. Their real-time audio distortion method used a pitch shifting algorithm to distort the speech. To hide the facial features they used facial detection, tracking and blocking algorithms. They tested their system on interview videos and demonstrated the system's ability to maintain the usability of the videos while protecting the identity of the people appearing in the videos.

Frederic Dufaux [14] discussed various Privacy Enabling Technologies (PET) in his work. He also evaluated all these PETs on the basis of how effective these PETs are in protecting privacy. Then he proposed a framework to determine the efficiency of different PET solutions to hide facial information and conceal identity. The results of his work showed the ineffectiveness of pixelization and blur techniques and the effectiveness of scrambling techniques. His work was a comparison of various techniques.

Sohn et al. [15] proposed a privacy protected video surveillance system that used JPEG extended range (JPEG XR). They used sub-band adaptive scrambling to obfuscate the privacy-sensitive face regions. They found that using sub-band adaptive scrambling on facial regions is more code efficient than using it on frames of the videos.

Saini et al. [1] proposed a method to determine the privacy loss value in a specific video. They suggested that privacy loss not only occurs from explicit channels like *who* information (people or facial information) but also from implicit channels such as, *what* (events), *when* (time), and *where* information (location) present in the videos. All of the above works except [1] worked on the principle of obscuring the facial or bodily features.

All of the above mentioned works except [1] only consider the explicit channels to prevent privacy loss. We mainly compare these related works based on the following aspects (Table 2.1): (1) whether the work obscures the explicit channel (who) to protect privacy, (2) whether the work considers the implicit channels of what, when, and where in calculating privacy loss, (3) whether the authors employ trust modeling, and (4) whether they consider the contextual knowledge in their proposed privacy protection method. While detecting and transforming human bounding boxes and facial regions may be necessary for preserving identity, it is generally not sufficient for protecting privacy. The adversaries can use the *what*, *when*, and *where* information present in the video, along with prior contextual knowledge, to infer the identities of individuals even without bodily cues. A model to measure the privacy loss from these implicit channels is proposed in [1]. In this work, it is assumed that the *when*, and *where* information can only be obtained from

Table 2.1: Previous works on privacy aware surveillance.

Work	**Hide Explicit Channels?**	**Hide Implicit Channels?**	**Trust Modeling?**	**Context Decoupling?**
Hudson and Smith [5]	Yes	No	No	No
Boyle et al. [6]	Yes	No	No	No
Barhm et al. [7]	Yes	No	No	No
Wickramasuriya et al. [8]	Yes	No	No	No
Zhang et al. [9]	Yes	No	No	No
Cheung et al. [10]	Yes	No	No	No
Kitahara et al. [11]	Yes	No	No	No
Fiadaleo et al [18]	Yes	No	No	No
Chen et al. [12]	Yes	No	No	No
Chaudhari et al. [13]	Yes	No	No	No
Dufaux [14]	Yes	No	No	No
Schiff et al. [19]	Yes	No	No	No
Sohn et al. [15]	Yes	No	No	No
Saini et al. [1]	No	Yes	No	No
Proposed work	No	Yes	Yes	Yes

the video. However, in traditional surveillance settings the operators have sufficient contextual information to obtain location and time information. Furthermore, the strong contextual knowledge of the operators increases the probability of them recognizing people in the video. In the proposed work, we propose to achieve the context decoupling by employing remote monitoring of surveillance videos and to the best of our knowledge, we are the first to propose such an anonymous surveillance framework. We also added an operator trust model to our system. This model is a reward/punishment based model to determine the credibility of an operator. The main idea of this trust model is adapted from [16] and [17]. A summary of the comparison of our work with earlier works is provided in Table 2.1.

2.2 Summary

This section briefly explained the available state-of-the-art surveillance systems, to give readers a glimpse of different privacy preserving techniques in the field of video surveillance. We also compare all the related works on the basis of following 4 factors:

- Whether the work obscures the explicit channel *who* to protect privacy.
- Whether the work considers implicit channels of *what*, *when*, and *where* information in calculating privacy loss.
- Whether the authors employ trust modeling.
- Whether they consider the contextual knowledge in their proposed privacy protection method.

Finally, we provided essential insights about the proposed anonymous surveillance framework.

Chapter 3

Proposed Anonymous Surveillance Framework

Traditional Closed Circuit Television (CCTV) based video surveillance systems are being replaced with automated digital systems. We can find surveillance systems in almost all public places, such as airports, schools, universities, banks and shopping malls. They are used to record and monitor the activities of people. While we are moving towards digitizing these systems, most are still used in the traditional way, where video streams are sent to a centralized control room at the local site. It has been observed that the number of cameras that an operator has to monitor in a traditional setup can be quite large. A study done by Dee and Velastin [20] suggested that the ratio of operators to cameras can be as low as 1:16. While performing the task of video monitoring, the visual attention of a security operator can drop below an acceptable level [21]. Wallace and Diffey [22] found that a security operator can only effectively monitor 4 camera views at a time. As there

are so many camera views, it is difficult to select any 4 camera views at a particular time. Atrey et al. [23] proposed a human-centric approach based system to select and schedule the four best camera views for an operator.

If an operator watches the same camera video for a long time, it might lead to leakage of the location information through too much familiarity with the place. The operator can search the web or describe the place to others to find location information. Once the location information is known, the adversary can establish a link between the remote site and the local site, leading to privacy loss. To combat this situation, we choose to do random re-assignment of the cameras after every quantum of time.

The rest of this chapter describes the proposed anonymous surveillance framework in detail. Section 3.1 presents an overview of the framework. The contextual decoupling strategy and the operator's trust model, which are essential components of the proposed framework, are described in Section 3.2 and Section 3.3, respectively. Finally, Section 3.4 summarizes the chapter.

3.1 An Overview of the Proposed Framework

We propose an anonymous surveillance framework that takes into account all of the above factors to show four different views to remote security operators. Although we plan to switch the views after a predetermined cut-off time, we should provide a means by which an operator can extend this view in case an important event is happening in the videos. However, we do not want them to misuse this feature, as extending a view many times means that the operators are watching the same videos and can develop context about

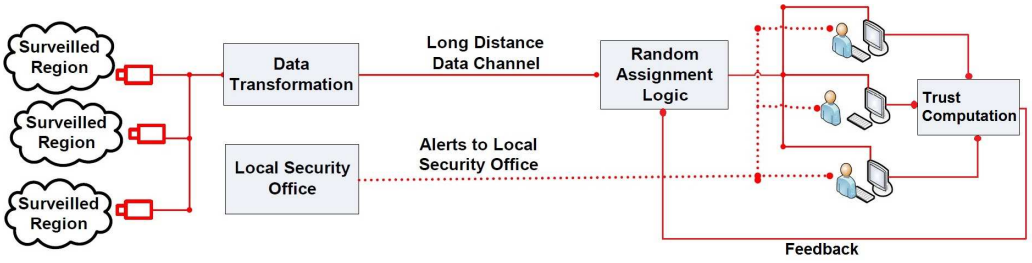


Figure 3.1: Proposed anonymous surveillance framework.

the monitored site which can lead to privacy loss. To overcome these two issues, we propose a context decoupling model as well as an operator trust model. The context decoupling model helps us in decoupling the context of an operator and determining the appropriate video switching time. The trust model, on the other hand, is a reward/punishment based model that assists us in verifying the credibility of an operator. To further strengthen our proposed framework, we monitor the trust level of every operator. If the trust level of an operator goes below a specific threshold, no videos are transmitted to that operator.

In the proposed framework, the video feeds are shown to the operators who are not familiar with the surveillance site so that removing text legends, logos, and other recognizable signs is enough to hide the where information. This is only possible if the video feeds are monitored remotely. Furthermore, the remote place must be chosen such that it has very low probability of contextual overlap with the surveillance site. Here context, can be defined over multiple dimensions, i.e. geographical, temporal, social, etc.

Figure 3.1 shows an overview of the proposed framework. Video camera feeds that are to be monitored remotely are first transformed to remove the

when and *where* information and are then sent over a long distance network to a distant place (ideally another continent). At the remote security office, these streams are randomly assigned to the operators. The operators at the remote office are given instructions regarding normal and abnormal situations, although no information about the monitored site is provided. In case of any abnormal situation, the viewers can anonymously signal the local security office. Consequently, the security personnel at the local office may access the surveillance site and take appropriate actions in real time immediately. The trust level is calculated for every operator and the feedback is sent to the random assignment block. If the trust level goes below a specific threshold, no videos are transmitted to that operator. In the following sections, we discuss the framework in greater detail.

3.2 Context Decoupling

The most important motivation behind the proposed anonymous surveillance framework is to decouple the local context from the video, i.e. the person who has knowledge of the local context where the camera is placed should not be given access to the video. Therefore, it is crucial to estimate the operator's contextual knowledge and to calculate the overlap between the local context and the operator's context. The first task is to develop criteria to decide how much contextual separation is sufficient for protecting privacy, and to develop a model for computing the contextual overlap. In the following subsections we discuss the core components of the contextual decoupling model.

3.2.1 Definition of local context

The context that enables a viewer to identify people in the video can span over multiple dimensions of (temporal, spatial, and cultural). Let $A = \{a_1, a_2, ...a_n\}$ be the set of contextual attributes that are important from a privacy perspective. These attributes can be spatial such as continent, country, city, premise; temporal such as absolute time, relative time; or cultural such as ethnicity, religion; etc.

3.2.2 Estimation of operator's context

There are two ways a person can obtain knowledge of the local context: i) from the video content and ii) through prior knowledge. We integrate these two aspects in our model as follows. Let $\mathcal{K}^q = \{k_1^q, k_2^q, ..k_n^q\}$ be the binary context vector that denotes what attribute the operator can identify from the video, and $\mathcal{K}^h = \{k_1^h, k_2^h, ..k_n^h\}$ is the binary context vector derived from the prior knowledge of the operator, which comes from his/her background. These context vectors are obtained using the following method:

$$k_i^q = \begin{cases} 1 & \text{if operator is able to obtain attribute } a_i \\ & \text{from the video;} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

and

$$k_i^h = \begin{cases} 1 & \text{if operator has history related to attribute } a_i; \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

The context vectors should be calculated for all the attributes that are identified as being important from a privacy perspective. In this work, we constructed these vectors by presenting questionnaires to the operators (as explained in Chapter 5).

3.2.3 Contextual overlap

With these vectors, the contextual overlap is calculated as:

$$\mathcal{C}_{opl} = \frac{\sum_{i=1}^{n}(\mathcal{K}_q \odot \mathcal{K}_h)}{A} \tag{3.3}$$

where $\odot$ is the element-wise OR operation. The current context increases if the person is watching the same video for a long time. Therefore, the probability of privacy loss $\mathcal{K}_{opl}$ is modeled as an exponential function of time:

$$\mathcal{K}_{opl} = 1 - e^{-(\mathcal{C}_{opl}+\epsilon)\times\mu\times t} \tag{3.4}$$

Here ϵ is a very small value which accounts for the fact that the operator can develop context by continuously watching the video even when the prior-contextual overlap is zero; and μ is a coefficient that depends on the unit chosen for time t. Note that the above formulation for computing the privacy loss $\mathcal{K}_{opl}$ is based on heuristics that the contextual knowledge increases exponentially with time [24].

Now, the operator o is eligible to monitor a video recorded at the given location if

$$\mathcal{K}_{opl} < \mathcal{K}_{max} \tag{3.5}$$

where $\mathcal{K}_{max}$ is specified based on the desired privacy level.

3.3 Operator Trust Model

As discussed in Section 3.1, we developed this framework in which the remote operators can watch the videos, send alerts to the local site if any abnormal event occurs in the videos, and also extend the viewing time if something important happens in the video so that they do not miss any important events. The videos are also switched randomly after a specified cut-off time. In our system, every time the videos switch, an operator can extend a video a maximum number of 5 times. This limit ensures that an operator is not extending the view unnecessarily and is a design choice. This trust model is based on the number of correctly and incorrectly identified events by an operator as well as the number of times s/he extended a view.

We all know that trust is an attribute that increases gradually with time, however you can lose trust rapidly even with a single mistake. In our model, we used this approach of gaining/losing trust and formulated an exponential model based on reward/punishment [16] [17]. The calculation of trust requires us to compute the reward and punishment for each operator. We need to compute trust for every operator to know how credible the operators are and whether they are performing their job efficiently. Trust level also helps us in identifying the surveillance effectiveness and privacy loss associated with each operator. Higher trust level indicates high surveillance effectiveness and suggests that operators are doing their job efficiently and causing no or little privacy loss. Low trust level indicates low observation adequacy and suggests

that operators are not performing their work effectively and increasing the risk of privacy loss. The details of calculation of reward, punishment and trust level are given in the following subsections:

3.3.1 Reward calculation

Reward is an attribute that increases slowly with time. Therefore, we calculate the reward using an equation which grows exponentially based on the number of correctly identified events by an operator and the number of extensions associated with each video. The equation to calculate the reward for an operator is as follows:

$$R_{(t)} = \frac{1}{d} \sum_{i=1}^{d} \frac{\alpha_r \times 2^{x_i}}{2^m} \tag{3.6}$$

where d is the number of correctly identified events by an operator, α_r is the parameter to control reward, x_i is the extension number for the i^{th} event in our system, m is the maximum number of extensions allowed, and $R_{(t)}$ is the value of the reward at the current time t. Note that we formulated an exponential equation for computing the reward $R_{(t)}$ on the basis of the heuristics that the trust level increases slowly with time [24].

3.3.2 Punishment calculation

Punishment is an attribute that increases quickly with time. The punishment calculation is done using an exponential function and it grows based on the number of incorrectly identified events by an operator and the number of extensions associated with each video. The equation to calculate the

punishment for an operator is given below:

$$P_{(t)} = \frac{1}{g} \times \sum_{i=1}^{g} \frac{\alpha_p \times 2^{x_i}}{2^m} \tag{3.7}$$

In the above equation, g is the number of incorrectly identified events by an operator, α_p is the parameter to control punishment, x_i and m are the same as in Equation 3.6 and $P_{(t)}$ is the value of the punishment at the current time t. The exponential equation for computing the punishment $P_{(t)}$ is based on the heuristics that the trust level decreases rapidly even if we make a single mistake [24].

3.3.3 Trust level calculation

For the final step of trust level calculation, we add the value of $R_{(t)}$ and subtract the value of $P_{(t)}$ from the previous trust level. We use the following equation to calculate the trust level of each operator:

$$\phi_{(t)} = \phi_{(t-w)} + b \times (R_{(t)} - P_{(t)}) \tag{3.8}$$

Where b is the weight given to the reward/punishment calculated at the time t, $\phi_{(t)}$ is the current trust level of an operator at the time t and $\phi_{(t-w)}$ is the previous trust value of an operator at $t - w$ time, w being the time window within which we recompute the trust level of operators.

Figure 3.2 shows the values of $R_{(t)}$ and $P_{(t)}$ for a single alert that a user generates, with respect to the number of extensions. Here, we can see that reward as well as punishment rises exponentially with respect to the number

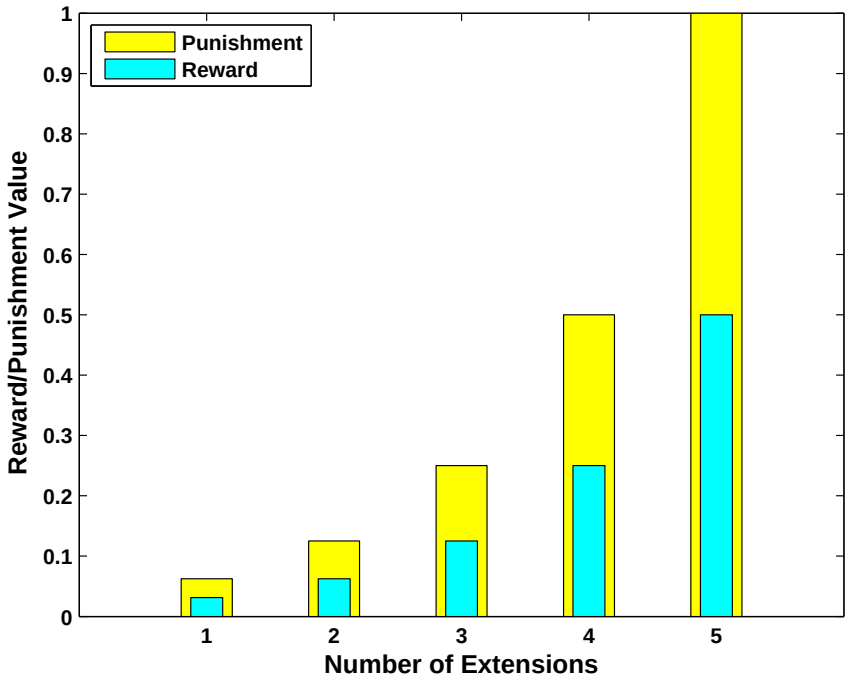


Figure 3.2: Reward/Punishment values versus number of extensions.

of extensions. Using equation 3.6 and Equation 3.7, we calculate both reward and punishment by varying the values of x_i from 1 to 5, keep the value of d and g as 1, the value of α_p as 1, the value of α_r as 0.5 and the value of m as 5.

3.4 Summary

In this chapter, we described the proposed anonymous surveillance framework and its various components such as context decoupling and operator trust model in detail. This forms the basis of i) the implementation of a system based on the proposed framework (described in Chapter 4) and ii) the experimentation and results analysis (presented in Chapter 5).

Chapter 4

System Implementation

To demonstrate the feasibility of the proposed anonymous surveillance framework we have implemented a system. The system is developed on a Windows PC using Visual Studio 2012 as the main IDE. C# is used as the primary coding language and the system is built on .net framework. MS SQL Server 2012 is also used as the database management system. This chapter describes the system details such as the dataset (in Section 4.1), the user interface and the basic data flow of the system (in Section 4.2). The chapter is summarized in Section 4.3.

4.1 Dataset

We used the Virat Video Dataset [25] as the primary source of surveillance videos to test the system. There were 329 different videos in the dataset that were combined to form 36 videos ranging in length from 10 minutes to 17 minutes each. We changed these 329 videos to 36 videos because the original

videos were too small to actually be used as surveillance videos. These 36 videos are further used to test the system as well as to perform experiments. The original format of the videos was MPEG-4 Part 14 commonly known as MP4. To create 36 videos from 329 videos, we had to join several videos together. We also changed the format to FLV as it is the most widely used format on WWW. We also developed an application in C# to prepare the ground truth for the dataset. The input for this application was the starting and ending frame numbers of all the events that happened in the video and the output was the starting and ending time of the events based on the play time of the video (by this we mean that our application played the video and kept increasing the frame numbers, and as soon as a counter matched a frame number associated with an event the time from the video was extracted). We used the AForge.NET library for this part of the work. AForge.NET is a computer vision and artificial intelligence library originally developed by Andrew Kirillov for the .NET Framework [26].

4.2 User Interface

As this is a web based application, different web pages were designed for the administrator and operators. When operators log-in to the application, they see 4 different videos and several radio buttons that are associated with each video. Using the radio buttons, operators send alerts to the local site, where the admin receives the alert as well as the camera (video name) and can act according to each alert, e.g. dispatching a guard, calling the police, etc. Remote operators also see an extend view button for each video with

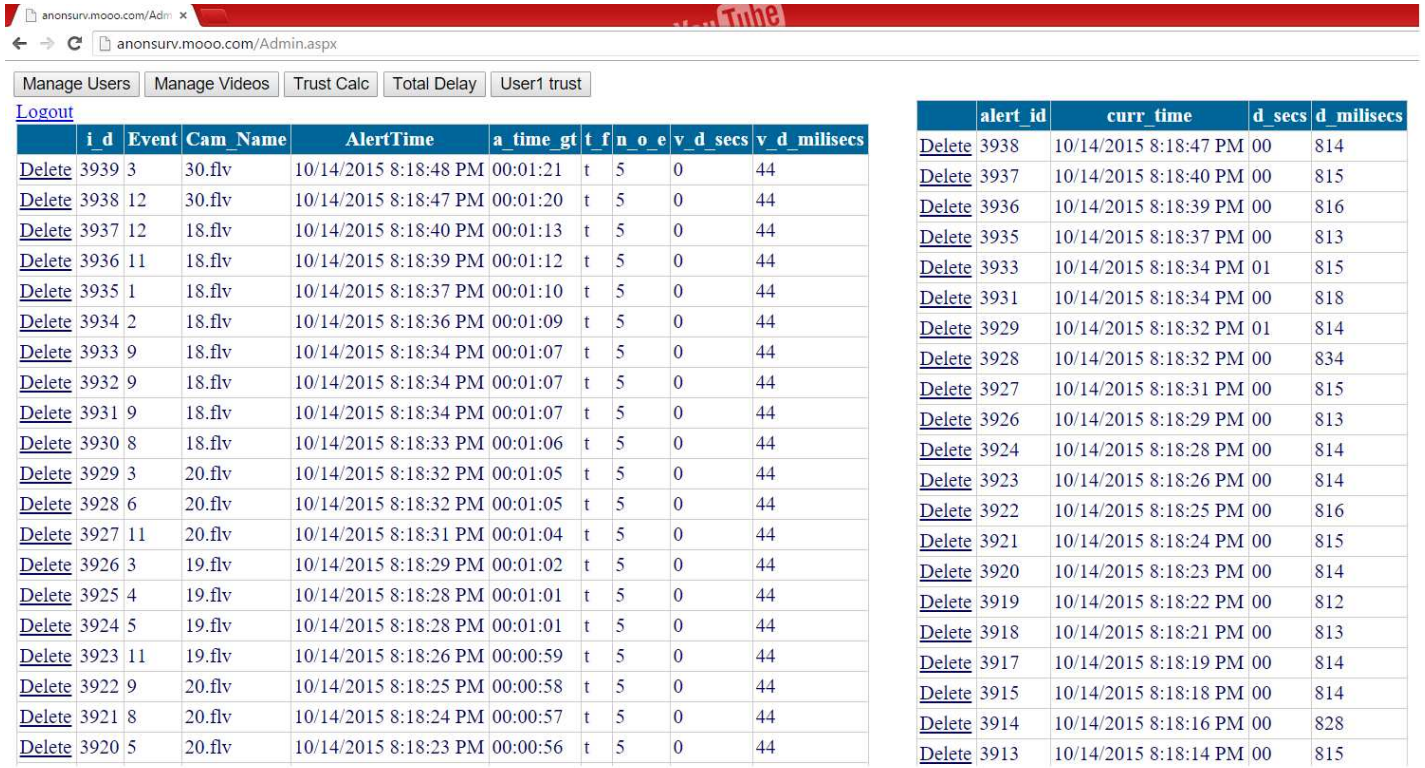
Table 4.1: Event types associated with the radio buttons on the operator's view.

Radio button number	**Event type**
1	Person loading an object to a vehicle
2	Person unloading an object from a vehicle
3	Person opening a vehicle/car trunk
4	Person closing a vehicle/car trunk
5	Person getting into a vehicle
6	Person getting out of a vehicle
7	Person gesturing
8	Person digging
9	Person carrying an object
10	Person running
11	Person entering a facility
12	Person exiting a facility

which they can extend the view multiple times. Recall that in our system, the maximum number of extensions that are allowed per video is 5 for every time the videos switch. All 4 videos are selected randomly from a database of videos and change after a specific cut-off time so that the operator's context is not developed over time.

On the local site, the admin accesses the application and sees the incoming alerts from the remote site. The admin also calculates the trust level of all the operators with the help of a trust calculation module which is designed based on the equations described in Section 3.3.

Figure 4.1 shows the administrator's view. An administrator can use the buttons shown to manage the users, manage videos (adding or deleting videos), calculate trust and check the total delay associated with each alert. In Figure 4.1, the table shown on the left is the alerts table that displays the alert id, event type, camera (video) name, date and time of the received



	i_d	Event	Cam_Name	AlertTime	a_time_gt	t_f	n_o_e	v_d_secs	v_d_milisecs
Delete	3939	3	30.flv	10/14/2015 8:18:48 PM	00:01:21	t	5	0	44
Delete	3938	12	30.flv	10/14/2015 8:18:47 PM	00:01:20	t	5	0	44
Delete	3937	12	18.flv	10/14/2015 8:18:40 PM	00:01:13	t	5	0	44
Delete	3936	11	18.flv	10/14/2015 8:18:39 PM	00:01:12	t	5	0	44
Delete	3935	1	18.flv	10/14/2015 8:18:37 PM	00:01:10	t	5	0	44
Delete	3934	2	18.flv	10/14/2015 8:18:36 PM	00:01:09	t	5	0	44
Delete	3933	9	18.flv	10/14/2015 8:18:34 PM	00:01:07	t	5	0	44
Delete	3932	9	18.flv	10/14/2015 8:18:34 PM	00:01:07	t	5	0	44
Delete	3931	9	18.flv	10/14/2015 8:18:34 PM	00:01:07	t	5	0	44
Delete	3930	8	18.flv	10/14/2015 8:18:33 PM	00:01:06	t	5	0	44
Delete	3929	3	20.flv	10/14/2015 8:18:32 PM	00:01:05	t	5	0	44
Delete	3928	6	20.flv	10/14/2015 8:18:32 PM	00:01:05	t	5	0	44
Delete	3927	11	20.flv	10/14/2015 8:18:31 PM	00:01:04	t	5	0	44
Delete	3926	3	19.flv	10/14/2015 8:18:29 PM	00:01:02	t	5	0	44
Delete	3925	4	19.flv	10/14/2015 8:18:28 PM	00:01:01	t	5	0	44
Delete	3924	5	19.flv	10/14/2015 8:18:28 PM	00:01:01	t	5	0	44
Delete	3923	11	19.flv	10/14/2015 8:18:26 PM	00:00:59	t	5	0	44
Delete	3922	9	20.flv	10/14/2015 8:18:25 PM	00:00:58	t	5	0	44
Delete	3921	8	20.flv	10/14/2015 8:18:24 PM	00:00:57	t	5	0	44
Delete	3920	5	20.flv	10/14/2015 8:18:23 PM	00:00:56	t	5	0	44

	alert_id	curr_time	d_secs	d_milisecs
Delete	3938	10/14/2015 8:18:47 PM	00	814
Delete	3937	10/14/2015 8:18:40 PM	00	815
Delete	3936	10/14/2015 8:18:39 PM	00	816
Delete	3935	10/14/2015 8:18:37 PM	00	813
Delete	3933	10/14/2015 8:18:34 PM	01	815
Delete	3931	10/14/2015 8:18:34 PM	00	818
Delete	3929	10/14/2015 8:18:32 PM	01	814
Delete	3928	10/14/2015 8:18:32 PM	00	834
Delete	3927	10/14/2015 8:18:31 PM	00	815
Delete	3926	10/14/2015 8:18:29 PM	00	813
Delete	3924	10/14/2015 8:18:28 PM	00	814
Delete	3923	10/14/2015 8:18:26 PM	00	814
Delete	3922	10/14/2015 8:18:25 PM	00	816
Delete	3921	10/14/2015 8:18:24 PM	00	815
Delete	3920	10/14/2015 8:18:23 PM	00	814
Delete	3919	10/14/2015 8:18:22 PM	00	812
Delete	3918	10/14/2015 8:18:21 PM	00	813
Delete	3917	10/14/2015 8:18:19 PM	00	814
Delete	3915	10/14/2015 8:18:18 PM	00	814
Delete	3914	10/14/2015 8:18:16 PM	00	828
Delete	3913	10/14/2015 8:18:14 PM	00	815

Figure 4.1: Administrator's view.

alert, whether the alert is true or false, the number of extensions and the video streaming delay in seconds and milliseconds. The table on the right is the alert delay table which shows the alert receiving delay in seconds and milliseconds. Figure 4.2 shows the operator's view of the application. Operators see 4 different extend view buttons that are associated with 4 videos. Radio buttons associated with each view are also shown to the operators and are numbered from 1 to 12. These radio buttons are basically the types of events that are happening in the videos. Table 4.1 shows the type of event associated with each radio button.

Basic system architecture is shown in Figure 4.3. It is shown in the figure that the admin and operator request the application using the internet, then they enter their credentials and log-in to the application. The request then goes to the application server. This is the server where the application is hosted using IIS (Internet Information Services). As the application has different web pages, the server validates the credentials, authenticates the users and assigns the correct web page on the basis of user role. On the local site the admin accesses the application in a web browser and see all the data stored in different database tables. Admin is not allowed to see the videos but manages all the user as well as video/camera data. Videos are stored in a database on a server on the local site. Whenever an operator accesses the application from the remote site, the flash videos are streamed over the internet using Hypertext Transfer Protocol (HTTP). We used HTTP as it is the most commonly used web protocol and almost all networks around the world allow incoming and outgoing HTTP requests.

All the visual components on the operator page are asp.net objects. The

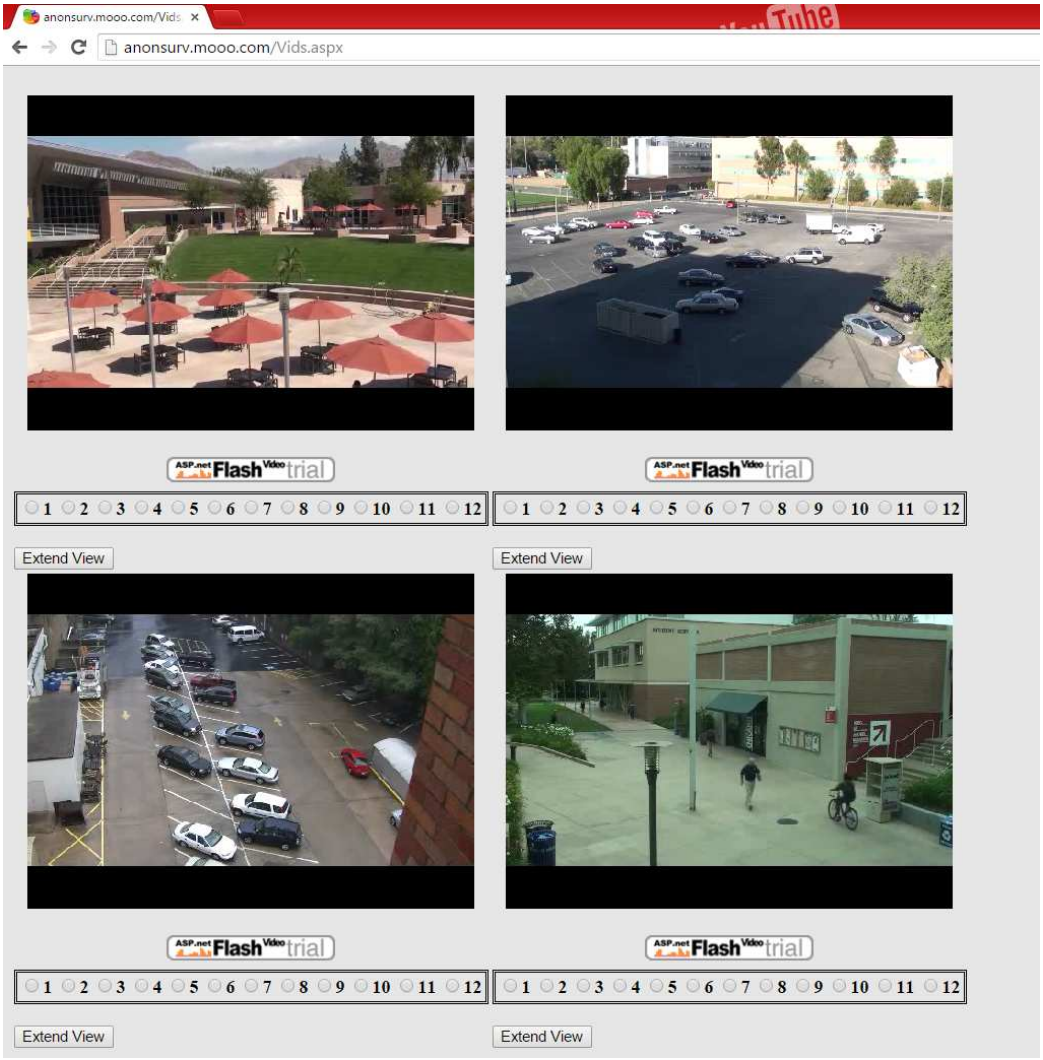


Figure 4.2: Operator's view.

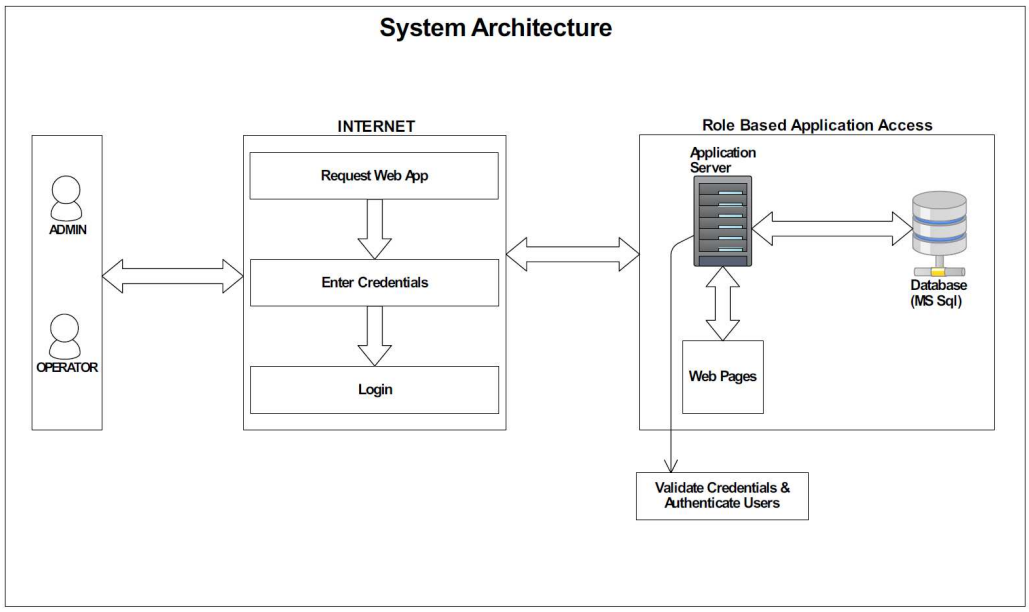


Figure 4.3: System architecture.

back end is coded in C#. As soon as a page is loaded, videos are streamed from the server to the operator's browser and the delay in streaming is stored in the label object using a timer. When the operator generates an alert using the radio buttons, new data is added to the database on the server. This request is completed using SQL queries. The data that is sent back to the server includes the operator name, current time, video name, event type, number of extensions for every video, and the video streaming delay time in seconds and milliseconds. On the local site, all these alerts are displayed to the admin so that the admin can take action on all the incoming alerts in real-time.

Figure 4.4 shows the data flow of the admin module. It shows all the web pages that an admin can access, as well as how the data is transmitted from these web pages to the database tables and vice-versa. Starting from the left

side, Figure 4.4 shows that an admin requests the log-in form. On the log-in form, the credentials are validated using the data saved in userdata table. After successful authentication the admin form is shown to the admin. The admin can now see the data from the alerts table and the alerts delay table. The admin can also check four other web pages that are specifically designed to manage the user data, manage the videos, perform the trust calculation and perform the total delay calculation.

On the manage users page, the admin can view, add and update data from the user data table. An admin can view, add and update data from the videos table on the manage videos page. On the total delay page, the admin can view and calculate the total delay associated with each alert using the data from the alerts and alerts delay table. The total delay is then stored in the total delay table. Using the trust calculation page, an admin can compute the trust level for every operator. First, the total number of correct and incorrect alerts for all the operators is computed using the data from the alerts table. This data is then stored in the total t/f alerts table. In the next step, reward and punishment is calculated using the data from the total t/f alerts table and the result is stored in the user_r_p table. Finally, the admin computes the trust level of every operator using the data from the user_r_p table and the result is stored in the trust level table.

Figure 4.5 shows the data flow of the operator module. Operators request the log-in form. The credentials of operators are validated using the data from the userdata table and the operator page is shown to the operators. As soon as this page is loaded, operators see 4 surveillance videos that are assigned randomly using the data from the video/cameras table. Operators

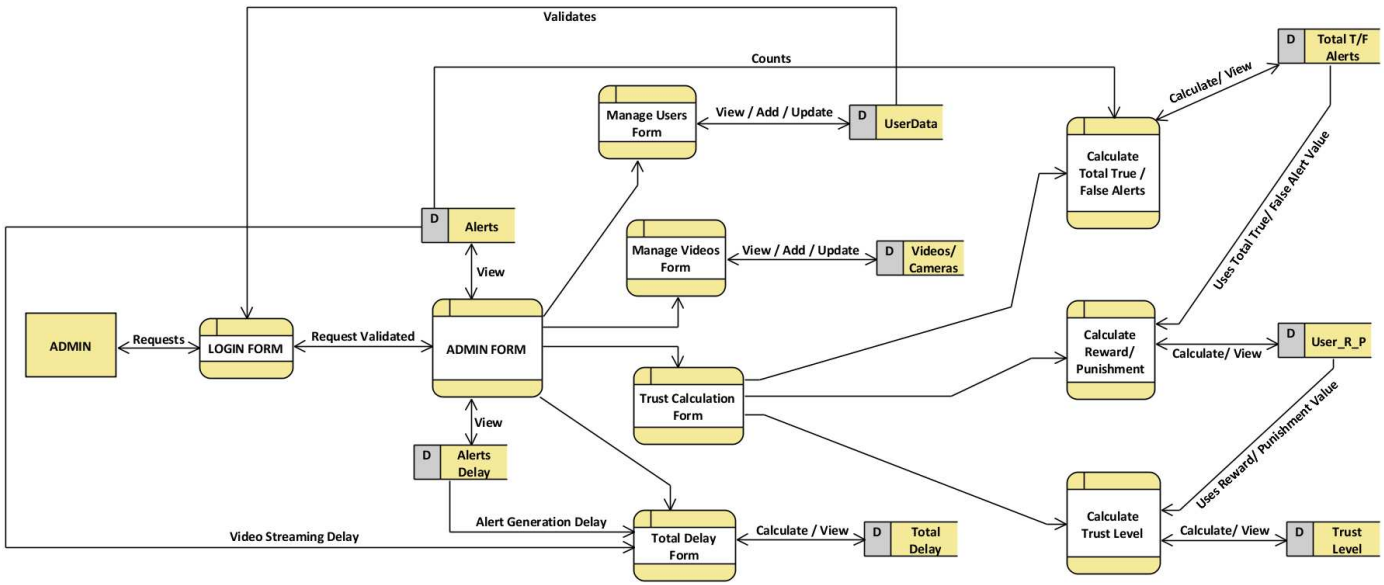


Figure 4.4: Admin data flow diagram.

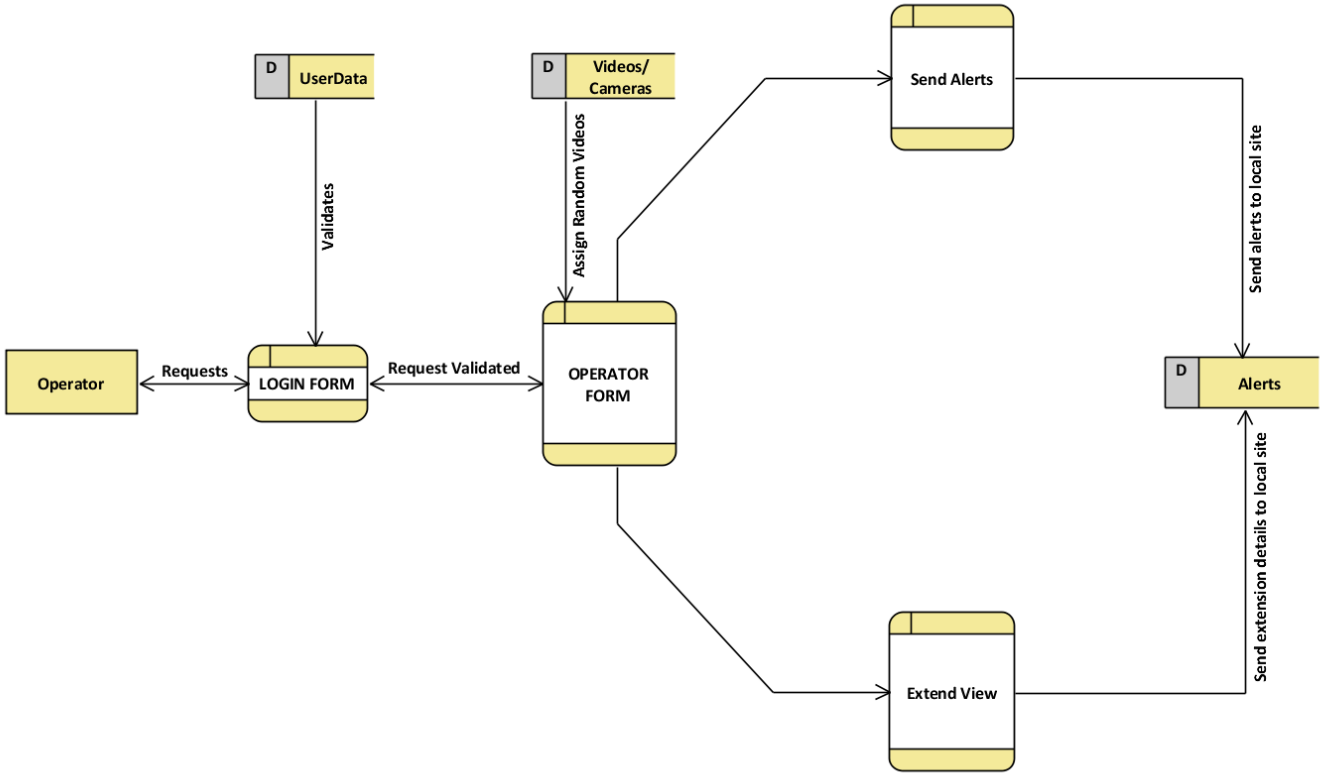


Figure 4.5: Operator data flow diagram.

can send alerts to the local site and extend the view. The alerts and extension data is stored in the alerts table.

4.3 Summary

All the details of the implemented system were discussed in this chapter. We discussed the operating system, platform, IDE and programming language used to develop the system. With the help of different figures, the admin and operator views of the developed system are also shown. We also discussed the working details of the application such as, video streaming, alert generation and how the application is hosted. Finally, we discussed the basic system architecture. Admin and operator data flow are also discussed in detail in this chapter.

Chapter 5

Experiments and Results

In this chapter we describe the experiments that we performed to test our system and present extensive results based on those experiments. The chapter is organized into five sections. In Section 5.1, we describe the experiments to determine the appropriate video switching time to ensure that the operator's contextual overlap is minimal. This uses the results of a user study based experiment, which were then validated using the contextual overlap model (described in Chapter 3, Section 3.2). Next, in Section 5.2, we performed a system test to check the surveillance accuracy and the surveillance real-timeliness. Then, in Section 5.3, we compute the trust level of the operators to determine their credibility using the trust model (described in Chapter 3, Section 3.3). Finally, the limitations of the proposed framework and the chapter summary are presented in Section 5.4 and Section 5.5.

5.1 Determining Appropriate Video Switching Time

5.1.1 User study based experiment

The main objective of this experiment is to find the appropriate video switching time for the proposed anonymous surveillance system. To perform this experiment, participants[1] were asked to log-in to an application, perform basic video surveillance tasks and generate alerts if they see anything suspicious in the videos. We used 10 of the 36 videos in the dataset for this experiment and participants were allowed to watch 4 random videos at a time out of these 10. There were 5 stages for this experiment and each stage was 10 minutes long. In the first stage, the cut-off time for video switching was one minute, for stage two it was 2 minutes and it kept increasing by one minute for each consecutive stage. At the end of every stage we asked 6 questions, given in Table 5.1, to all the participants. We calculated the surveillance accuracy (number of correctly identified events/ total number of events) from the user generated alerts, and the privacy preservation accuracy (percentage of participants that suggested there is no privacy loss) from the questionnaire data for all 5 stages, and selected the appropriate video switching time where privacy preservation accuracy and surveillance accuracy were at a maximum.

Figure 5.1(a) shows the percentage of 'yes' answers for Q1, and Figure

[1]As this experiment involved human participants, appropriate ethics approval was obtained from the University Human Research Ethics Board (UHREB).

[2]Privacy loss is the measure of the who, where, when and what aspects of the information that can be gained from the video data [1]

[3]Utility loss of the video data refers to the decrease in the degree of accuracy with which security tasks can be accomplished.

Table 5.1: Questionnaire for user study based experiment

No.	Question
Q1	Were you able to recognize any person in the videos you just finished watching?
Q2	Were you able to recognize any place in the videos you just finished watching?
Q3	Were you able to identify the time in the videos you just finished watching?
Q4	As per the above definition of privacy loss [2], do you think any privacy loss has occurred in the Videos you just finished watching? Rate between 1(no privacy loss) and 5(full privacy loss)?
Q5	As per the above definition of utility loss [3], do you think any utility loss has occurred in videos you just finished watching? Rate between 1(no utility loss) and 5(full utility loss)?
Q6	While you were watching the videos they kept switching after a specific cut-off time. Rate between 1(least) to 5(most) on the basis of how suitable was the video switching time to efficiently perform surveillance tasks?

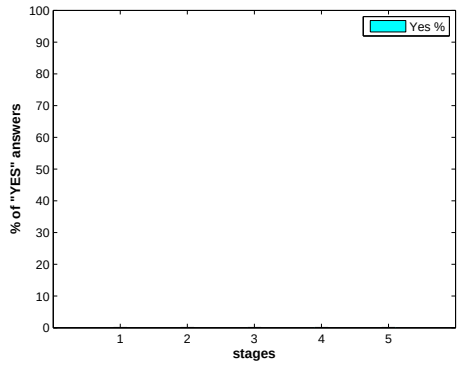
5.1(b) shows the percentage of 'yes' answers for Q2. The results of Q1 suggested that 100% of the participants were not able to identify people, and the results of Q2 were similar to Q1, as no participant was able to identify the location. Figure 5.1(c) shows the percentage of 'yes' answers for Q3. For Q3, we got the best response for stages 1, 2 and 3, as no more than 30% of participants were able to identify the time. Figure 5.1(d) shows the rating of privacy and utility loss for Q4 and Q5 respectively. As a response to Q4 and Q5, 90% of the participants suggested that there is no privacy loss for stage two whereas 60% of the participants suggested no utility loss in stages 2 and 3. Figure 5.1(e) shows the rating of the most suitable switching time for Q6. It was found that 80% of the users rated stages 2 and 3 as the most suitable time for video switching. Finally, Figure 5.1(f) shows the surveillance accuracy for every stage of the experiment. The surveillance accuracy for every stage was calculated by finding the sum of correctly identified alerts generated by each participant and dividing it by the total number of alerts generated in that stage. From this figure, it can be seen that surveillance

accuracy was higher in stage 2 and stage 3 compared to other stages.

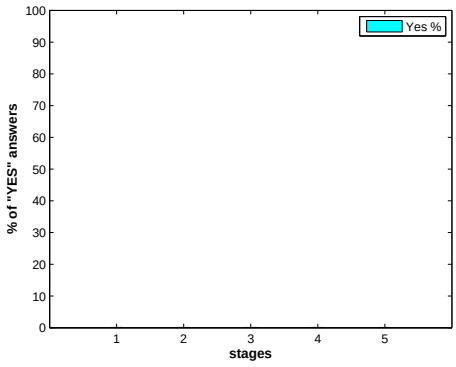
After carefully examining the results of all 5 stages, it was found that every other stage had some drawbacks in comparison to stage 2. Participants found more utility loss in stage 1, and more privacy loss in stages 3, 4 and 5 as compared to stage 2. As per the results of the user study, stage 2 was the optimal choice for the appropriate video switching time. We used the results from this experiment to determine the contextual overlap of the participants. The process of evaluating the contextual overlap is explained in the next subsection.

5.1.2 Calculation of contextual overlap

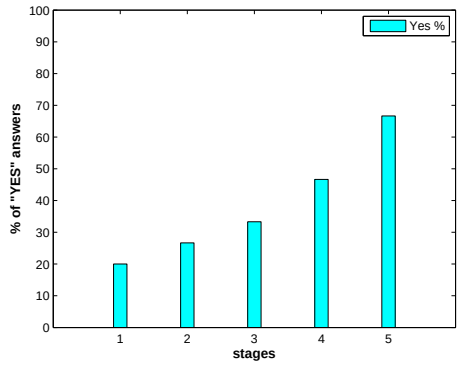
The results of every stage of the user study based experiment in the previous subsection suggested that the appropriate video switching time should be 2 minutes so that the operator's contextual knowledge about the local site and people in the video is not developed. To further validate our results, we used the context decoupling method described in Section 3.2.3 to determine the contextual overlap of every user that participated in the experiment. To calculate the value of contextual overlap we used questions Q1, Q2 and Q3 as the three attributes. Then we used Equation 3.1, defined in Section 3.2.2, to create the binary vectors on the basis of user responses to these three questions. We also created another binary vector based on equation 3.2 in Section 3.2.2, but as no participant has the history associated with the location, all the values were zero in this vector. Using these two binary vectors, we calculated the value of contextual overlap ($\mathcal{C}_{opl}$) using equation



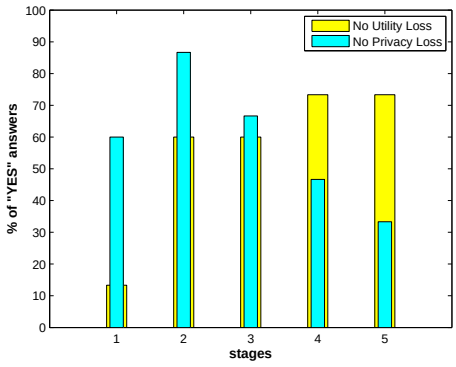
(a) Results of Q1



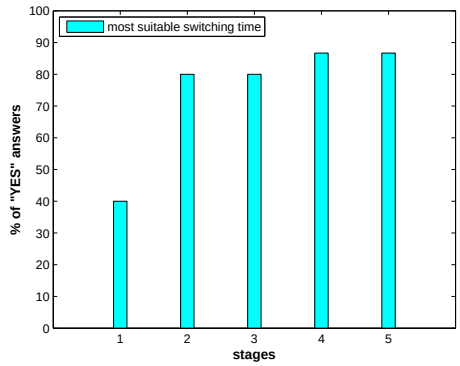
(b) Results of Q2



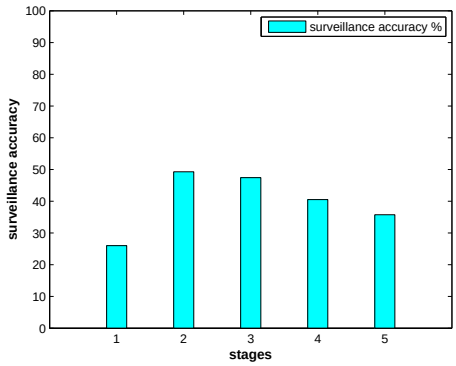
(c) Results of Q3



(d) Results of Q4 and Q5

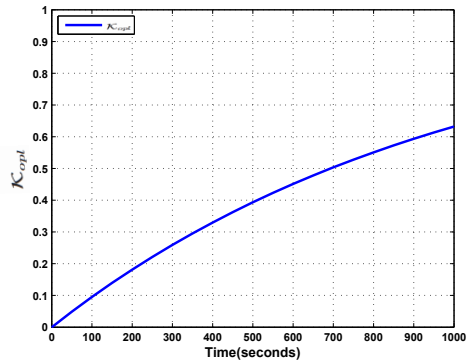


(e) Results of Q6

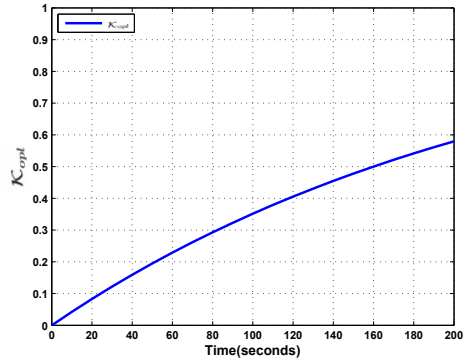


(f) Surveillance Accuracy

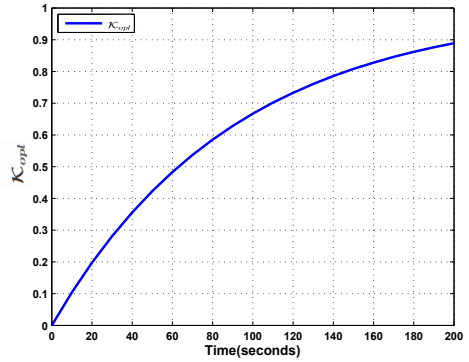
Figure 5.1: Results of questionnaire and surveillance accuracy for all 5 stages.



(a) Privacy loss probability for zero contextual overlap; $\epsilon = 0.1$; $\mu = 0.01$.



(b) Privacy loss probability for contextual overlap = 0.333; $\epsilon = 0.1$; $\mu = 0.01$.



(c) Privacy loss probability for contextual overlap = 1; $\epsilon = 0.1$; $\mu = 0.01$.

Figure 5.2: Privacy loss probability for contextual overlap.

3.3. The minimum value of contextual overlap C_{opl} for 15 participants was 0 and the maximum value was 0.333.

The privacy loss varies based on both contextual overlap as well as the time. It increases with the length of time during which the operator watches a particular camera video continuously. We have kept $K_{max} = 0.5$ so that the probability of privacy loss is less than half. Figure 5.2(a) shows the current context value with time when the prior-contextual overlap is zero. We see that even when the prior-contextual overlap is zero, the privacy loss goes beyond 0.5 if the operator watches same camera continuously for a long time (11 minutes 40 seconds in the given case). However, it is almost impossible to find operators with zero contextual overlap. As discussed above, the maximum value of C_{opl} for all the participants was 0.333. With this value of C_{opl}, we draw the values of K_{opl} in Figure 5.2(b). We can see that in this case, cameras need to be switched more quickly as the privacy loss goes beyond 0.5 after only 2 minutes 42 seconds.

Finally, the privacy loss probability for a participant with maximum contextual overlap is plotted in Figure 5.2(c). Because of the large contextual knowledge of this participant, the privacy loss probability goes beyond the threshold after just 64 seconds. Switching the cameras so frequently may reduce the effectiveness of the surveillance operator and therefore it is not a desirable solution. From this experiment, we conclude that in the case of remote users, the contextual overlap increases slowly, and therefore we only need to switch the videos after a reasonable amount of time, which is practical. Experiments in Section 5.1.2 suggest that appropriate video switching time should be 2 minutes 42 seconds, which also matches the results of Section

5.1.1. Note that automatic event detection techniques can be also integrated to add a criterion for the camera re-assignment strategy [23], which would mean that cameras cannot be switched in the middle of an event.

Furthermore, the proposed framework recommends a large cloud of remote operators monitoring cameras around the globe, similar to outsourced call centers. The size of the operator pool should be large enough for the random re-assignment to be effective in context hiding. The periodic re-assignment of the cameras also increases the effectiveness of the operators by reducing the dullness and fatigue that comes from monitoring the same video over a long period of time [27]. However, the cameras should be assigned such that every operator gets to observe almost an equal number of targets. As the amount of attention (or work) required from an operator is proportional to the number of the targets in the camera view, a workload equalizing method can be used for camera to operator assignment [28].

5.2 System Testing for Surveillance Accuracy and Timeliness

The purpose of this experiment is to test the prototype system for surveillance accuracy and real-timeliness. Participants were asked to log-in to the system, perform surveillance tasks, generate alerts and extend the view if necessary. Surveillance accuracy is determined based on the ratio of the number of correctly identified events compared to the total number of events. There were two types of delays associated with this test: 1) video streaming delay,

Table 5.2: Event types merged into groups

Radio button	**Event groups**
1	Entering/exiting a facility
2	Entering/exiting a car
3	Putting things in a Car
4	Running/gesturing

and 2) alert response delay (delay between the time the event is generated on the local site and the time the alert comes back the local site). The total delay is calculated by adding up these two delays. Surveillance real-timeliness is determined based on the total delay. We aim to keep this delay as small as possible in order to say that our system works in real time. We used appropriate the video switching time determined in Section 5.1.2 as the cut-off time to switch videos for this experiment. A total of 15 users participated in this experiment and they sent alerts to the local site. Total time for this experiment was 30 minutes. They marked their responses for 12 different event types described in table 4.1. Participants were able to correctly identify 366 events out of 681 total events. The surveillance accuracy for the system testing using the appropriate video switching time was 53.74%.

As this accuracy was low, we began to investigate the reason. Consequently, we planned to redo this experiment by merging 12 event types in 4 groups. This time we used 4 radio buttons instead of 12. Table 5.2 show 4 groups that we used for radio buttons. Now, participants marked their responses based on these 4 groups. More importantly, this time the four event groups were shown on the interface with the event descriptions rather than the event numbers. Figure 5.3 shows the operator view for this experiment. A total of 15 participants took part in this experiment and they were able to

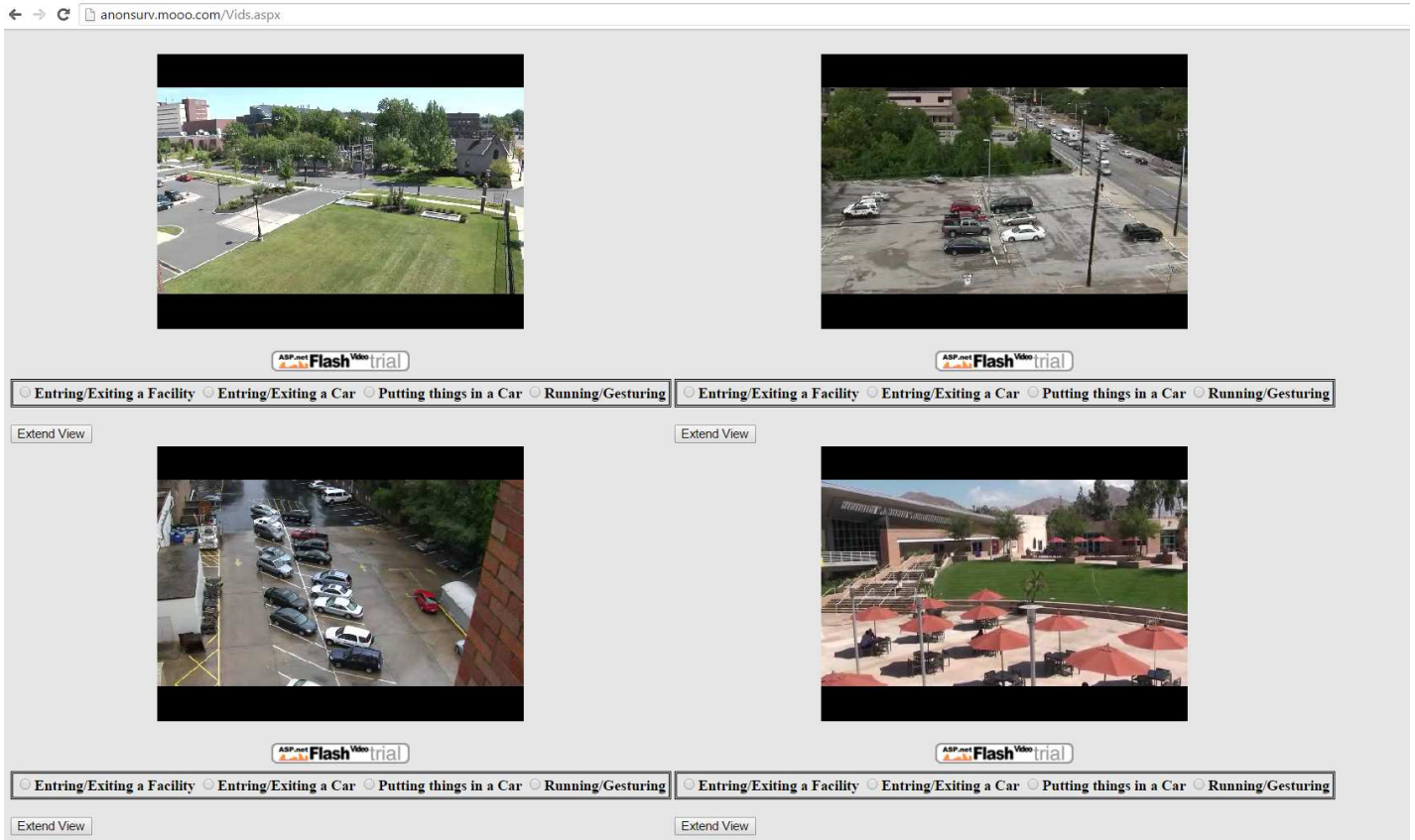


Figure 5.3: Revised operator's view showing event descriptions instead of event numbers.

correctly identify 831 events out of 1127 total events. Surveillance accuracy increased almost 20% as the accuracy for this experiment was 73.73%. From this experiment it was found that the operators were more receptive to event descriptions and remembering events by numbers was difficult for them, that was what the reason was for the low accuracy in the previous experiment. We believe that with an improved and more friendly user interface that reduces operators' fatigue can further help improving this accuracy.

All the alerts generated by the participants, along with the the video streaming delay is shown in figure 5.4(a). It can be clearly seen from figure 5.4(a) that the video streaming delay from the local to the remote site is just few milliseconds. Figure 5.4(b) also shows the delay for sending an alert from the remote site to the local site with respect to all the alerts that participants generated. This delay never exceeded 2 seconds. Finally, the total delay is calculated by adding up both the above-mentioned delays and is shown in figure 5.5. The total delay always remained below 2 seconds, which means that the admin at the local site can work on alerts in real-time.

5.3 Trust Computation

The purpose of this experiment is to calculate the trust level of the operators for determining their credibility. As discussed earlier, in our system the participants marked their responses using the radio buttons and they were sent to the local site. The admin on the local site pressed a button to calculate the trust level (evaluate credibility) of every participant at the same time. Using equation 3.8 (given in Section 3.3), the value of trust (i.e. $\phi_{(t)}$) is

Manage Users | Manage Videos | Trust Calc | Total Delay | User1 trust

Logout

Alert ID	Event Type	Camera Name	True/False	Number of Extensions	Streaming Delay in Seconds	Streaming Delay in Milliseconds
3871	9	29.flv	t	1	0	6
3870	7	12.flv	t	1	0	6
3869	4	29.flv	t	1	0	6
3868	9	12.flv	t	1	0	6
3867	1	29.flv	t	1	0	6
3866	9	29.flv	t	1	0	6
3865	5	29.flv	t	1	0	6
3864	9	29.flv	t	1	0	6
3863	5	29.flv	t	1	0	6
3862	7	12.flv	t	1	0	6
3861	6	29.flv	t	1	0	6
3860	9	12.flv	t	1	0	6
3859	4	28.flv	t	4	0	14
3858	5	28.flv	t	4	0	14
3857	4	11.flv	f	4	0	14
3856	6	11.flv	f	4	0	14
3855	5	11.flv	f	4	0	14
3854	6	11.flv	f	4	0	14
3853	6	28.flv	t	4	0	14

1 2 3 4 5 6 7 8 9 10 ...

(a) Alerts with video streaming delay

Alert ID	Alert Delay in Seconds	Alert Delay in Millieconds
3871	00	692
3870	00	628
3869	00	416
3868	00	667
3867	00	940
3866	00	694
3865	01	5
3864	01	300
3863	00	455
3862	01	33
3861	00	75
3860	01	11
3859	00	660
3858	01	181
3857	00	381
3856	00	167
3855	00	610
3854	00	484
3853	00	741

1 2 3 4 5 6 7 8 9 10 ...

(b) Delay in sending alerts from remote to local site

Figure 5.4: Video streaming delay and the delay in sending alerts from the remote site to the local site.

Alert ID	Total Dealy in Seconds	Total Delay in Milliseconds
3871	0	698
3870	0	634
3869	0	422
3868	0	673
3867	0	946
3866	0	700
3865	1	11
3864	1	306
3863	0	461
3862	1	39
3861	0	81
3860	1	17
3859	0	674
3858	1	195
3857	0	395
3856	0	181
3855	0	624
1 2 3 4 5 6 7 8 9 10 ...		

Figure 5.5: Total delay

calculated for every operator and is saved in a database table.

As trust level can only vary from 0 to 1, to calculate the trust level of participants the initial trust is kept at 0.5 so that all the participants start from the same level of trust. The total number of correct and incorrect alerts are calculated for every participant with respect to the number of extensions. These values are then stored in a SQL database. $R_{(t)}$ and $P_{(t)}$ are then calculated for every participant using the values from the database with the help of equations 3.6 and 3.7, which are explained in sections 3.3.1 and 3.3.2 respectively. These values are also stored in a database table. These stored values of $R_{(t)}$ and $P_{(t)}$ are used to calculate the trust level of every participant using equation 3.8, explained in section 3.3.3.

To show the specific instances when the trust level increased, we generated

User Name	True/False	Total Value	Number of Extensions
user1	t	13	1
user1	t	2	2
user1	t	10	3
user1	t	9	4
user1	t	30	5
user1	f	0	1
user1	f	0	2
user1	f	0	3
user1	f	0	4
user1	f	0	5

User Name	True/False	Value
user1	t	0.297
user1	f	0

User Name	Trust Level
user1	0.619

Figure 5.6: Increased trust level

User Name	True/False	Total Value	Number of Extensions
user1	t	0	1
user1	t	0	2
user1	t	0	3
user1	t	0	4
user1	t	0	5
user1	f	13	1
user1	f	2	2
user1	f	10	3
user1	f	9	4
user1	f	30	5

User Name	True/False	Value
user1	t	0
user1	f	0.595

User Name	Trust Level
user1	0.262

Figure 5.7: Decreased trust level

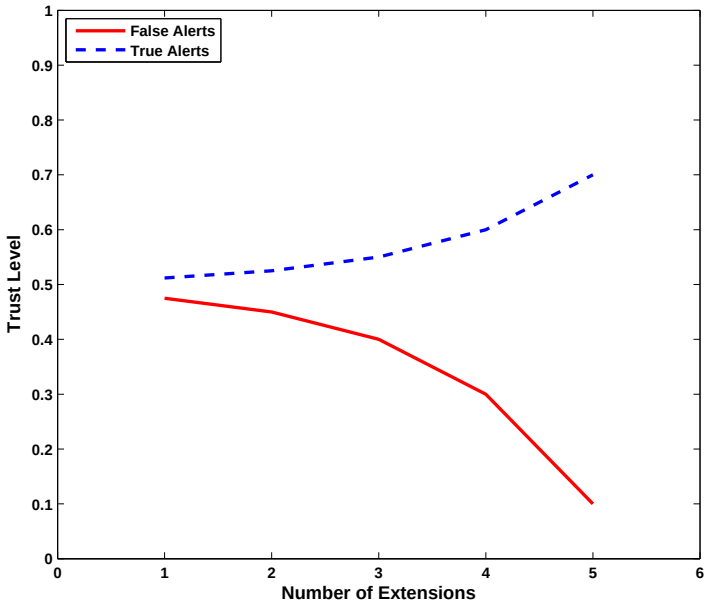


Figure 5.8: Evolution of trust with respect to number of extensions

64 alerts for an operator named user1. We randomly increased the number of extensions for every alert generated and then explicitly marked all events as correct. The left table in Figure 5.6 shows the total number of alerts generated by user1 with respect to the number of extensions. The center table in Figure 5.6 shows the total value of reward $R_{(t)}$ and punishment $P_{(t)}$ calculated with the values shown in the left table using equation 3.6 and equation 3.7. The right table in Figure 5.6 shows the increased value of the trust level, which was calculated using equation 3.8 and the values displayed in the center table. To show the decrease in the trust level we used the same approach but changed all the alerts to incorrect. Figure 5.7 shows the decreased trust level for the same user.

To show the evolution of the trust level with respect to the number of extensions, we calculated trust level values by explicitly marking all the alerts

as correct and increasing the number of extensions from 1 to 5. We repeated the same process by marking all the alerts as incorrect. Figure 5.8 shows the evolution of trust level with respect to the number of extensions for a single user. It is clear from figure 5.8 that if all the alerts are correct and the total number of alerts is kept the same, the trust level increases with the increasing number of extensions. However, if all the alerts are incorrect and the total number of alerts is kept same, the trust level decreases with an increasing number of extensions. Furthermore, it can be seen that the trust level increases slowly, but the decrease is much faster.

We also computed the trust level using the value of $R_{(t)}$ from Figure 5.6, which is 0.297, and the value of $P_{(t)}$ from Figure 5.7, which is 0.595. The final trust level is computed using equation 3.8. The previous trust value was kept at 0.5. The final trust value after this calculation was 0.3808. This clearly shows that if an operator identifies more events incorrectly, the trust value will decrease.

Another test was done to show the trust level evolution of user1. For this test, user1 started at a trust level of 0.5 and performed the surveillance tasks using the developed system. We calculated his trust after every 10 minutes for 5 subsequent stages. Figure 5.9 shows the evolution of trust for user1 over the period of 50 minutes, as each stage was 10 minutes long.

The trust level value can be used as feedback to the developed system, and other features based on this trust level can be added to the system. In the future, we plan to use the trust level and extend the system to a point where no videos will be streamed to the operators if the trust level falls below a specific threshold. We can also add a feature where videos from sensitive

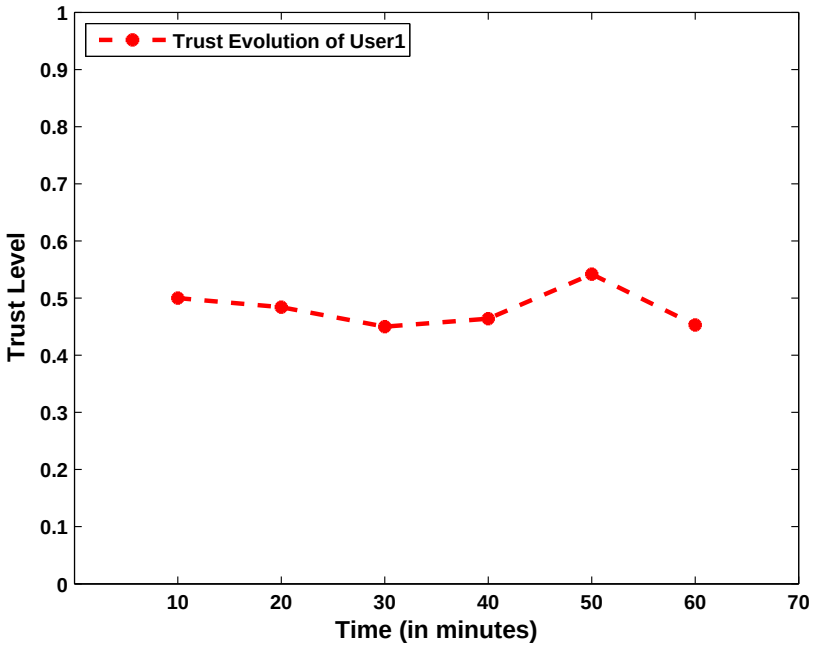


Figure 5.9: Evolution of trust for user1

locations like schools, banks, hospitals, airports, etc. will only be streamed to operators with a high trust level.

5.4 Limitations

There are a few limitations of the proposed anonymous surveillance framework. First, the proposed framework requires a large pool of cameras and of operators to make the video switching effective. Second, it is always a challenge to find operators with zero or little contextual knowledge. From a system implementation perspective, we used HTTP protocol to stream the videos, which sends out a single stream of data to a client's web browser and does not take into account the quality of the internet connection/speed at

the client end. We could have used RTMP (Real Time Messaging Protocol), which has many advantages over HTTP such as sending out multiple streams and combining them on the client's end. It also automatically changes the quality of the stream based on the client's internet speed.

5.5 Summary

In this chapter, we discussed various experiments and their results. First, we performed a user study based experiment followed by the calculation of contextual overlap, which helped us determine the appropriate video switching time at the operator's end. Next, we performed a thorough system test to evaluate the surveillance accuracy and timeliness. This experiment showed that the proposed framework is feasible to implement and the resulting system meets the required timeliness. Finally, we showed the results of the proposed trust computation model and its evolution. In addition to the experimental results, we also provided the limitations of the developed anonymous surveillance framework.

Chapter 6

Conclusion and Future Work

It is very difficult to provide robust privacy preservation in traditional surveillance systems. The operators usually have a substantial prior knowledge of the surveillance site and the persons depicted therein, which can cause significant privacy loss even in the absence of the bodily cues. In order to reduce this privacy loss, the contextual knowledge of the operator needs to be decoupled from the surveillance site.

Based on the above-mentioned idea, this thesis proposes an anonymous surveillance framework, in which surveillance videos are monitored remotely by operators who are kept unaware of the location of the camera/videos. We stream the videos to a remote location which is sufficiently far away from the site being monitored. Even if we take these measures, it is still difficult to find operators with zero contextual knowledge. If these operators watch the same videos for a long period of time, they can learn more about the people and site which leads to privacy loss. To overcome this issue, we changed the videos after a specific cut-off time, which was determined using

experiments. We also implemented and tested the system for feasibility issues such as operator surveillance accuracy and timeliness. An operator trust model which is based on a reward/punishment strategy is also introduced to determine the credibility of the operators.

In the future we plan to use an adaptive quality control mechanism to selectively transmit video data, as the transmission of videos to a remote place requires a large bandwidth. As the video data travels over the Internet it passes through many untrusted networks. Therefore, we can apply data encryption before transmitting the data. We can use the data encryption methods discussed by Carrillo et al. [29]. We can also incorporate some other methods of data security discussed in [30] in the system for better reliability.

We also plan to use RTMP for streaming videos as it is more flexible and secure as compared to HTTP. We aim to extend the trust model, use the trust level as feedback to the system and stop streaming videos to the operators if their trust level falls below a specific threshold. We can also add another module to the system where videos from sensitive locations such as, airports, schools, hospitals, etc., are only streamed to the operators with a high trust level. We can also introduce a workload equalizing method as described by Saini et al. [28] for assigning cameras to the operators.

Appendix A

Preliminaries

In order to develop the proposed anonymous surveillance system, we developed prototypes in different languages. One of the first prototypes that we developed was written in C++ using sockets to stream the video data from server to client. For the next step, we moved to Asp.net and used C# to create this application. We chose ASP because the system being developed was a web based system and ASP had a very large collection of libraries and other wrapper classes that were required for the development of this project. We needed a solid computer vision library to process the video data to be used for the project. We worked with different libraries but ultimately selected AForge.NET because it had all the necessary features required for our work.

At the start of the project we requested video data from the University of Winnipeg's surveillance repository but later we decided to use the Virat video dataset because it was publicly available. We also had to decide which video format to use for streaming the videos, as the original videos in the

Virat dataset were in MP4 format. After evaluating different video formats we selected flash video (FLV) as the final format for video streaming. We chose FLV because it was easy to stream videos from server to client in ASP as FLV is the most commonly used video format for web-based streaming. We also needed to choose between HTTP and RTMP for video streaming as both the protocols have some advantages over the other. We decided to use HTTP over RTMP because sometimes RTMP streams are blocked by the firewall. These were the basic building blocks of the proposed system. In the following sections, Section A.1 to Section A.5, we give basic details about the dataset, video format, protocols and the libraries that we used to build the system.

A.1 Virat Video Dataset

The Virat video dataset [25] is a large-scale real world surveillance video dataset. Sangmin et al. [25] released the dataset in October 2011. It was supported by DARPA (Defense Advanced Research Projects Agency) and the videos were shot at different locations across the USA. It is designed to be realistic, as they recorded the videos in public places with a natural cluttered background. The dataset contains both ground and aerial videos in high resolution as well as downsampled versions. A number of people and vehicles appear in different videos, and they annotated 12 different event types described in Table 4.1. The dataset is freely available to the public and can be downloaded from MIDAS. We used ground videos to test our system and there were a total of 329 different videos in the dataset.

A.2 FLV

Flash Video (FLV) [31] is a data wrapper for sending video over the Internet. It was originally developed by Macromedia. We used FLV for our project as this is the most widely used video format on the WWW. We had to convert the original MP4 videos to FLV using the available video converting software such as Movavi video converter, Allok video converter, etc.

A.3 HTTP

Hypertext transfer protocol [32], commonly known as HTTP, is the underlying protocol used by various applications on the WWW. It is the most used protocol on the web and is the foundation of all the communication on the WWW. For our project, we used FLV videos and streamed them to a client PC over the internet using HTTP. HTTP was developed by Tim Berners-Lee and his teammates at the European Organization for Nuclear Research (CERN) in 1989. HTTP works on Transmission Control Protocol (TCP) port number 80 or 8080 and is the most cost effective and simple protocol to host and stream flash videos. The advantages of using HTTP for streaming flash video is that almost no internet connection blocks incoming data on port 80. Thus, the content is streamed to the client without any hassle. HTTP streaming is known as progressive download video, which means that if you are watching a 50 minute video and you want to skip to the last 10 minutes of the video, you have to wait until the whole video is downloaded to the browser's cache. That is why today, streaming websites such as YouTube,

HULU, VEVO, etc. use RTMP to stream videos. We used HTTP because a surveillance video is being streamed live and there is no need to skip forward or backward during the real world scenario of video surveillance.

A.4 RTMP

RTMP is the real time messaging protocol [33], initially developed by Macromedia to stream video, audio and data between a Flash Player and a server over the Internet. RTMP also uses TCP for streaming the audio and video data. There are different versions of the protocol, mainly RTMPS, RTMPE and RTMPT. RTMPS stands for secure RTMP and utilizes a TLS/SSL connection to stream the content. RTMPE stands for enhanced RTMP. It uses the Diffie-Hellman Key exchange and HMACSHA256 to generate a pair of RC4 keys, one of which is used to encrypt the data going out of the server while the other is used for the encryption of incoming data. RTMPT uses tunneling of an RTMP stream over HTTP to bypass any corporate firewalls. Hence, T stands for tunneling. RTMP streaming has many advantages over HTTP streaming. It is more secure as compared to HTTP streaming. Clients can also view a RTMP streamed video at any point and it is not saved in the browser's cache.

A.5 AForge.NET

AForge.net is an open source computer vision and artificial intelligence framework [26]. It was developed by Andrew Kirillov in 2006. It is a framework

developed for C# .net and it consists of different libraries. Some of the libraries included in the package are imaging, video, vision, machine learning, etc. We used AForge.net to create the ground truth for our dataset. Some of the built-in functions in video and vision libraries helped us in achieving the goal of ground truth construction.

Bibliography

[1] M. Saini, P.K. Atrey, S. Mehrotra, S. Emmanuel, and M.S. Kankanhalli. Privacy modeling in video data publication. In *IEEE International Conference on Multimedia & Expo*, pages 60–65, Singapore, 2010.

[2] M. Saini, P. Atrey, S. Mehrotra, and M.S. Kankanhalli. W3-privacy: Understanding what, when, and where inference channels in multi-camera surveillance video. *Multimedia Tools and Applications*, 68(1):135–158, 2014.

[3] M. Saini, P.K. Atrey, S. Mehrotra, , and M.S. Kankanhalli. Anonymous surveillance. In *IEEE International Conference on Multimedia & Expo*, pages 1–6, Barcelona, Spain, 2011.

[4] D. Klitou. Backscatter body scanners-a strip search by other means. *Computer Law & Security Report*, 24(4):316–325, 2008.

[5] S.E. Hudson and I. Smith. Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. In *ACM Conference on Computer Supported Cooperative Work*, pages 248–257, Boston, MA, USA, 1996.

[6] M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. In *ACM Conference on Computer Supported Cooperative Work*, pages 1–10, New York, NY, USA, 2000. ACM.

[7] M. Barhm, N. Qwasmi, F. Qureshi, and K. el Khatib. Negotiating privacy preferences in video surveillance systems. In *Modern Approaches in Applied Intelligence*, pages 511–521, Syracuse, NY, USA, 2011. Springer.

[8] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian. Privacy protecting data collection in media spaces. In *ACM International Conference on Multimedia*, pages 48–55, New York, NY, USA, 2004. ACM.

[9] W. Zhang, S. Cheung, and M. Chen. Hiding privacy information in video surveillance system. In *IEEE International Conference on Image Processing*, volume 3, pages II–868–71, Genoa, Italy, 2005.

[10] S. Cheung, M.V. Venkatesh, J.K. Paruchuri, J. Zhao, and T. Nguyen. Protecting and managing privacy information in video surveillance systems. In *Protecting Privacy in Video Surveillance*, pages 11–33. Springer London, 2009.

[11] I. Kitahara, K. Kogure, and N. Hagita. Stealth vision for protecting privacy. In *International Conference on Pattern Recognition*, pages 404–407, Cambridge, U.K, 2004.

[12] D. Chen, Y. Chang, R. Yan, and J. Yang. Tools for protecting the privacy of specific individuals in video. *EURASIP Journal on Applied Signal Processing*, 2007(1):107–107, 2007.

[13] J. Chaudhari, S. Cheung, and M.V. Venkatesh. Privacy protection for life-log video. In *IEEE Workshop on Signal Processing Applications for Public Security and Forensics.*, pages 1–5, Los Alamitos, CA, USA, 2007.

[14] F. Dufaux. Video scrambling for privacy protection in video surveillance: recent results and validation framework. *Society of Photo-Optical Instrumentation Engineers Conference Series*, 8063(2), 2011.

[15] H. Sohn, W. De Neve, and Y.M. Ro. Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in jpeg xr. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(2):170–177, 2011.

[16] C.A. Bhatt, P.K. Atrey, and M.S. Kankanhalli. A reward-and-punishment-based approach for concept detection using adaptive ontology rules. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 9(2):10:1–10:21, May 2013.

[17] M.A. Hossain, P.K. Atrey, and A. El Saddik. Learning multisensor confidence using a reward-and-punishment mechanism. *IEEE Transactions on Instrumentation and Measurement*, 58(5):1525–1534, 2009.

[18] D.A. Fidaleo, H.A. Nguyen, and M. Trivedi. The networked sensor tapestry (nest): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In *ACM International Workshop on Video Surveillance & Sensor Networks*, pages 46–53, Washington, DC, USA, 2004.

[19] J. Schiff, M. Meingast, D.K. Mulligan, S. Sastry, and K. Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. *Protecting Privacy in Video Surveillance*, pages 65–89, 2009.

[20] H.M. Dee and S.A. Velastin. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, 19(5-6):329–343, 2008.

[21] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *IEEE Signal Processing Magazine*, 22(2):38–51, 2005.

[22] E. Wallace and C. Diffley. *CCTV control room ergonomics. Technical Report 14/98, Police Scientific Development Branch (PSDB), UK Home Office*, 1988.

[23] P.K. Atrey, A. El Saddik, and M.S. Kankanhalli. Effective multimedia surveillance using a human-centric approach. *Multimedia Tools and Applications*, 51(2):697–721, 2011.

[24] J.B. Oommen and L. Rueda. A formal analysis of why heuristic functions work. *The Artificial Intelligence Journal*, 164:1–22, 2005.

[25] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.C. Chen, J.T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan,

J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3153–3160, Washington, DC, USA, 2011. IEEE Computer Society.

[26] A. Kirillov. Aforge.NET. `http://www.aforgenet.com/`, 2006. [Online; accessed 21-December-2006].

[27] M. Saini. *Privacy-Aware Surveillance System Design*. PhD thesis, National University of Singapore, 2012.

[28] M. Saini, X. Wang, P.K. Atrey, and M.S. Kankanhalli. Adaptive workload equalization in multi-camera surveillance systems. *IEEE Transactions on Multimedia*, 14(3):555–562, 2012.

[29] P. Carrillo, H. Kalva, and S. Magliveras. Compression independent object encryption for ensuring privacy in video surveillance. In *IEEE International Conference on Multimedia & Expo*, pages 273–276, Hanover, Germany, 2008.

[30] T. Winkler and B. Rinner. Securing embedded smart cameras with trusted computing. *EURASIP Journal on Wireless Communications and Networking*, (8), 2011.

[31] Macromedia. FLV. `https://goo.gl/l1TYkW`.

[32] Tim Berners-Lee. HTTP. `https://goo.gl/T01Nbi`, 1989.

[33] Macromedia. RTMP. `http://goo.gl/gS1BD8`.