



uOttawa

L'Université canadienne  
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES  
ET POSTDOCTORALES**



**uOttawa**

L'Université canadienne  
Canada's university

**FACULTY OF GRADUATE AND  
POSTDOCTORAL STUDIES**

**Bassem Barake**

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

**M.A.Sc. (Electrical Engineering)**

GRADE / DEGREE

**School of Information Technology and Engineering**

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

**Towards an Intelligent Surveillance System for Public Security at Crowded Places**

TITRE DE LA THÈSE / TITLE OF THESIS

**Prof. El Saddik**

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

**Prof. J. Zhao**

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

**EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS**

**Prof. W. Gueaieb**

**Prof. F. Yu**

**Gary W. Slater**

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

# Towards an Intelligent Surveillance System for Public Security at Crowded Places

by

Bassem Barake

Thesis submitted to the  
Faculty of Graduate and Postdoctoral Studies  
In partial fulfillment of the requirements  
For the M.Sc. degree in  
Electrical and Computer Engineering

School of Information Technology and Engineering  
Faculty of Engineering  
University of Ottawa



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-59892-4  
*Our file* *Notre référence*  
ISBN: 978-0-494-59892-4

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

## Abstract

Nowadays, safety and security of public areas has become the centre of attention, and especially after 9/11 attack. A visual surveillance for public security is proposed to meet the needs of public areas, such as shopping malls, train stations, airports, etc...

Surveillance researchers from the computer vision have focused on building automated systems and have hardly adopted an approach, where human is involved in decision making with the help of computer inputs. Knowing that the fully automated systems have some advantages such as saving human power, and supporting remote monitoring, but in other hand, these systems are not often tended to work in complex environments. This thesis adopts a *human-cum-machine* centric approach, where this approach is followed in order to assist security personnel in the physical monitoring with the support of video surveillance. A proposed system is designed and developed, which provides security personnel with real-time information about any suspected person they want to suspect about. Using the security system's application, the officer has to identify the region within the surveilled area where the suspected person is located. Based on the region identified, the system delegates and controls nearest installed cameras to cover the region and then capture pictures of the scene. Once the suspected person is identified by the security personnel from the pictures captured, the corresponding information such as personal and contact information of the person is displayed and available to the security officers. The architecture of the system is based on web services technology, where many researches outlined the benefits of using web services in terms of scalability, reliability, and re-usability of its components. This thesis provides detailed information about the proposed system and its components, discusses the application's graphical user interface to be used by security personnel, and proposes a methodology to select and control installed security cameras to cover a region identified by security personnel.

## **Acknowledgements**

I would like to thank many people who have kindly contributed to this Master study. First, I would like to express my thorough gratitude to my supervisors, Dr. Abdulmotaleb El Saddik and Dr. Jiying Zhao for their support, assistance, and encouragement. I would like also to express my gratitude to Dr. Pradeep Atrey for his time, valuable comments, suggestions and criticism which have helped me to present this work. I would like to express my deep gratitude to all my friends for the support, understanding, comfort and encouragement. Finally, I want to thank every member of my family and especially my mother Sana, for her invaluable support, I would like also to thank my sisters Rana and Rola, my niece Sana, my nephews Karim and Jad, my brother-in-law Mazen, and my grandmother Ifdal.

Thanks to all of you.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Studies</b>	<b>7</b>
2.1	Surveillance System Architecture . . . . .	7
2.2	Camera Control . . . . .	9
2.3	Human Centric Approach . . . . .	11
<b>3</b>	<b>Proposed System and Methodology</b>	<b>14</b>
3.1	System Overview . . . . .	14
3.1.1	Human block . . . . .	16
3.1.2	Machine block . . . . .	17
3.2	Proposed Architecture . . . . .	18
3.2.1	Multimedia web-services . . . . .	19
3.2.2	Intelligent decision service . . . . .	22
3.2.3	Application's Graphical User Interface . . . . .	26
3.3	<i>Human-cum-machine</i> Centric Approach . . . . .	28
3.3.1	User's Roles . . . . .	28
3.4	Camera Control . . . . .	30
3.4.1	Determining possible angles to cover a subregion in 2-D . . . . .	31

3.4.2	Determining a camera's panning angle, and panning direction . . .	37
<b>4</b>	<b>Proposed GUI, and System Implementation</b>	<b>40</b>
4.1	Application's graphical user interface . . . . .	40
4.1.1	Camera View UI Component . . . . .	41
4.1.2	Map View UI Component . . . . .	42
4.1.3	Personal Info View Component . . . . .	46
4.1.4	Scene picture UI Component . . . . .	48
4.2	System implementation overview . . . . .	49
4.2.1	Client Application . . . . .	50
4.2.2	Web service components . . . . .	51
<b>5</b>	<b>System Performance Evaluation and Results</b>	<b>62</b>
5.1	Graphical User Interface Evaluation . . . . .	62
5.1.1	Fixed Layout display Approach (FLDA) . . . . .	63
5.1.2	Dynamic Layout display Approach (DLDA) . . . . .	66
5.1.3	Comparison between FLDA and DLDA approaches . . . . .	67
5.2	System Response Time . . . . .	68
5.2.1	Overall User Satisfaction . . . . .	70
5.3	Camera Control Evaluation . . . . .	73
5.3.1	Experiment 1: Evaluating Cameras' Effort . . . . .	74
5.3.2	Experiment 2: Quality of Scene's Pictures . . . . .	76
5.3.3	Comparison to fuzzy logic approach camera control . . . . .	78
<b>6</b>	<b>Conclusion and Future Work</b>	<b>81</b>
<b>7</b>	<b>Appendix A</b>	<b>83</b>

# List of Tables

5.1	Overview of participants . . . . .	63
5.2	Average time taken to identify a sub region in FLDA . . . . .	64
5.3	Average time taken to identify a sub region in DLDA . . . . .	66
5.4	System Response Time . . . . .	69
5.5	Overall System usability . . . . .	71
5.6	Dynamic vs Fixed Layout Approach . . . . .	72
5.7	Percentage of images that identified the target taken within three attempts	77
5.8	Input test data categories . . . . .	80

# List of Figures

1.1	Flow chart . . . . .	3
3.1	Block Diagram . . . . .	15
3.2	High Level System Architecture . . . . .	20
3.3	Adding sensor and landmrak to the system . . . . .	22
3.4	Retrieving personal information . . . . .	24
3.5	Delegating and controlling list of cameras . . . . .	25
3.6	Layout generation and presentation . . . . .	27
3.7	UML Use Case Diagram . . . . .	29
3.8	2D plan view showing initial camera's state . . . . .	32
3.9	2D plan view showing the region to cover . . . . .	35
3.10	2D plan view showing the new panning angles to cover the specified region	36
4.1	Officer in the hallway . . . . .	44
4.2	Dynamic Layout display: Subregions representation . . . . .	44
4.3	Dynamic Layout display: Rotated map to be displayed on the GUI . . . .	45
4.4	Fixed Layout display: Subregions representation . . . . .	46
4.5	Fixed Layout display: Rotated map to be displayed on the GUI . . . . .	47
4.6	Screenshot of the system application . . . . .	49

5.1	Without using landmarks vs Using landmarks - Fixed layout approach .	65
5.2	Without using landmarks vs Using landmarks - Dynamic layout approach	67
5.3	Laboratory setup . . . . .	74
5.4	Average efforts of the three installed cameras . . . . .	75
5.5	Subregion Coverage Percentage . . . . .	76

# Chapter 1

## Introduction

Nowadays, due to the rise in various terrorist threats, security of public places such as shopping malls, banks, airports, train stations and subways, has been seen as an important issue [5, 38, 15, 18, 27]. In order to prevent these threats, security agencies usually exercise several measures; which include physical monitoring by the security personnel [22] and the use of technological means such as automated video surveillance [2], [16]. Both of these approaches have some advantages and limitations.

In the case of physical monitoring, security officers can make a self decision based on their observation and experience while they are patrolling the area and/or screening the people/baggage for dangerous goods, metal and non-metallic objects [4]. Although this approach is effective, it consumes a lot of time, efforts, and is often expensive. Automated video surveillance approach tends to solve or reduce the limitations of the physical monitoring approach since it saves the man power and also supports remote monitoring. However, on the other hand, the current video surveillance technologies often tends to work in constrained environments that needs special considerations and setups. Moreover, developing a fully automated video surveillance system that can work in complex environments (e.g. outdoor), is still under research. Having known the

limitations of the two approaches, it is desirable to follow an approach that combines the advantages of the two approaches and reduces their limitations.

This thesis follows a *human-cum-machine* centric approach and proposes to develop a system to address the limitation in actual procedures. The proposed system assists a security personnel in the physical monitoring with the aid of video surveillance technology. Unlike a fully automated system, the proposed system allows the users (i.e. security personnel) to get involved in the decision making process by utilizing the video surveillance technology (e.g. face detection, person identification, etc.) in a semi-automated way. The rationale behind our *human-cum-machine* centric approach is based on the envision that “computers can assist but cannot replace humans” [25].

Our proposed system has potential use in a typical scenario of security officer patrolling a public area. The scenario is described using the flow chart shown in figure 1. We assume that the security officer possesses a handheld device (e.g. laptop, PDA, etc). When the security officer suspects a person in the crowded area, the application running on the handheld device provides the background information about the person. This is done using the following steps. When the security officer suspects a person, s/he selects from the layout displayed on the handheld device a region in which the suspected person is. Once the region has been selected by the security officer, the system alerts multiple cameras installed in the area to maximize the coverage of the specified region. Thereafter, the system starts sending pictures of the scene containing the suspected person. Several pictures of the scene are displayed on the handheld device, out of which the security officer selects the one which is of satisfactory quality and contains the face image of the suspected person. This face image is then processed to extract the features in order to identify the person. The extracted features are sent to the central server for matching and the matched results are received back on the handheld device. Based on

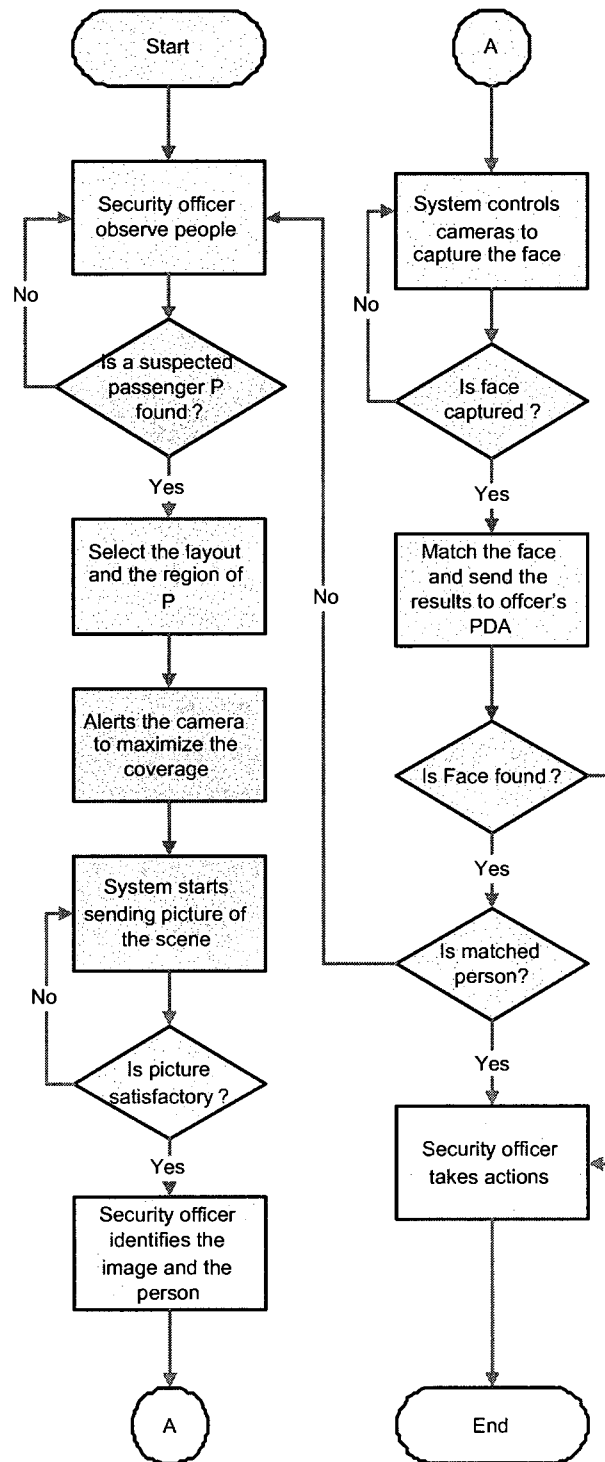


Figure 1.1: Flow chart

the obtained results, the security officer takes the appropriate actions such as calling the central station, or inputting additional information about the person. In case the face feature could not be extracted from the image selected, the system triggers the cameras to track the identified person to capture a good image.

The realization of the proposed system has multiple research challenges. We divide them into four major categories and describe as follows:

1. *System Architecture*: Scalability is an important concern, which is the ability to expand (for example: support more users) without the impact on the overall performance of the system. System extensibility is also an issue, where it allows the system to be extended by adding new components or features over the period of time, or replacing any existing component without disturbing any existing code. Another requirement that the system could have, is to allow the connection and interaction between the client and server regardless of the platform and computer programming language they are using. Therefore, designing a system architecture that meet the above requirements is an issue.
2. *GUI on a Portable Device*: The portable devices such as PDA held by the security officers usually have small screens with a limited display space. Designing a user interface for these kind of devices is challenging, since it is required to give the user the full functionality of the application with a user friendly interface. In the proposed system, as the security officer uses GUI in order to input the location of the suspected person via a mouse or a pen, it could be difficult for the officer to provide the location of the person when the supervised area is wide and large. The user interface must address this issue by providing the user with some help in identifying the location of the subject.
3. *Camera Control*: Maximizing the coverage of a certain area is another issue. When

the location of the suspected person is identified by the officer, the cameras are needed to be controlled in a way that they can cover the location and capture the images of the suspected person. The methodology should take into consideration the initial parameters, location, and the position of each camera.

4. *Face Matching*: When a good face image of the suspected person is captured, it is sent to the central server for matching with the faces stored in the database. The face matching, which requires facial feature extraction and comparison, has been extensively addressed by the face recognition research community.

Note that, in this thesis, we focus on three issues - System Architecture, Application Graphical User Interface, and Camera Control; while we use VeriLook library for face detection and identification. In summary, our contributions corresponding to the above three issues can be stated as follows:

- The proposed system adopts a Web Service technology based architecture, which provides a simplified mechanism that connects application regardless on the technology or devices that is used by the officer.
- A human centric approach has been followed in order to assist the security officer, who plays an important role in the security surveillance system. Unlike the fully automated systems, the aim of the proposed system is to get the officer involved and to be assisted for the better accomplishment of the security objectives. We designed and developed a system that meets the above requirements.
- We also propose a novel methodology to select and control the cameras in order to cover the region where the a suspected person is identified by the security officer.

The rest of the thesis is organized as follows. Chapter 2 describes the related studies, followed by chapter 3 which describes the proposed system architecture, and the

camera control strategy for controlling the surveilled cameras. Chapter 4 presents the application's graphical user interface (GUI), and discusses the system implementation, and chapter 5 evaluates the system performance, usability of the application's GUI, and the camera control strategy followed. Finally, chapter 6 concludes the thesis with a discussion on possible future research directions.

# Chapter 2

## Related Studies

In this chapter we present a review of the recent contributions made to the field of surveillance from the following three different aspects - Surveillance system architecture, Human centric approach, and Camera control strategy. The past works related to these three aspects are described in the following.

### 2.1 Surveillance System Architecture

As reported in [33], the maritime port surveillance presented in [23] is based on a client/server architecture design. This kind of design is tied up to set of programming languages, and the services offered could only be used within the party offering and developed the services. However, our system is based on web services technologies, which is a simplified mechanism that connects application regardless of the technology or devices that is used, and allows the use of existing web services that is offered and developed by other party regardless of the implementation language they use.

PRISMATICA (pro-active integrated systems for security management by technological institutional and communication assistance) [34, 33], ADVISOR (annotated digital

video for intelligent surveillance and optimised retrieval) [1, 33], and surveillance systems [24, 19] are all based on *distributed architecture* where this kind of architecture allows multiple user and application to share resources, such as cameras or sensors. However, these systems have not explored the concept of web services technology. Introducing the proposed web service architecture into the surveillance systems allows the functionality to be extended over a period of time.

Few applications have adopted web services technology. In [29], an IBM smart surveillance system has been presented, which is an automatic video analytics that generates real time alerts based on the generated event from sensors, and allows users to search through these events. System management, storing events as well as retrieving these events from the database, are done using web services technologies. Although our system architecture is on the similar lines to that of [29], we propose an architecture that gives user more privileges and complex functionalities to use towards the system. In addition to that, we also present web service components that provide some business logics such as selection security cameras, and controlling cameras. The IBM surveillance system is an automated system, in which, the alerts are generated in real time based on events triggered, but in our case security officer manually triggers events for camera selection and control.

Information retrieval for digital libraries in [12] is based on the adaptation of web services technology where the benefit of using web services is the re-usability of its components. For instance in [12], the information related to users query is offered to the requestor which could be reused and accessed by other web services or components whenever needed.

UBILearn system in [8] is a web based application where the author had presented performance measurement of the system. Since it is also based on the adaptation of

web services technology, changing the presentation tier from web based application to a desktop application has no impact on the scalability and overall system performance. The author's test scenario was based on sending a SOAP request to a service which handles the SOAP request, retrieve data from a database, and return the response back in SOAP format to the client. The author ran multiple concurrent tests in order to perform the scalability and performance under two different circumstances 100 Mb/s-Based LAN and DSL cable. The tests show that web services-based application performs and scales well.

Many researches such as in [13, 11, 26] outlined the benefits of using web services. As defined in [11], web services are modular where their components could be used by other components, available to any components need to use them, based on SOAP over HTTP that are able to be invoked over the internet, which defines a common standard allowing the system to be interoperated, defined, and implementation-independent.

Finally, we believe that services-oriented vision offers many benefits to our system, where we proposes intelligent decision web services that to be accessed by other web services or application's component. It also allows integrating or replacing any existing modules to an application instead of redeveloping them. For instance, the camera control and selection strategy proposed and implemented in our system, could be later replaced by any service that provides better results without obstructing any existing code.

## 2.2 Camera Control

There has been a lot of work in the area of camera control. Hence, we describe the ones that are closely related to our work, and highlight how our strategy is different from the existing ones.

Determining a region of interest in a classroom captured by a panoramic camera has been presented in [39]. The region of interest, which is tend to be the region containing

the lecturer, is cropped from the panoramic video that would result in a ‘virtual camera’ that can be controlled by panning, tilting, and zooming. The approach presented could not be applied in our proposed system, since the authors took advantage of having the lecturer always in the camera view. In our case, people could be moving from a region to another and that would result in having a person in a camera view and then moved to another camera field view.

In [17], a method for camera control has been proposed so that each the field of view of each camera in the network does not overlap with another view. However in our proposed system, field of view overlapping from different cameras is not a concern, since the system allows the cameras to take pictures for the same region, so that several pictures are available for security officer to choose from.

In [30, 31], a cooperative approach has been presented where cameras compete and cooperate to provide a more efficient surveillance. Cameras are assigned with tasks to detect a facial image of an intruder. The winner camera of the task is controlled to track the face of the intruder, and the other cameras continue to look for other facial images or other region of interest in the area.

In our previous work [4], a ‘Camera Control’ approach was proposed using fuzzy logic. The system uses fuzzy inference system that emulates human decision behaviour. It reasons in the same manner as humans do when it comes to deciding on which camera is better to select, and in which direction and angle it should pan to take a good image of a passenger. The FIS interprets fuzzy input variables to determine fuzzy output variables needed to activate cameras and control it to take image of a passenger. However, in our proposed system, security officer is assigned the task to manually select the region of interest based on his/her observation, and near cameras are automatically controlled in a way to cover the region selected.

Previous contributions have proposed some camera control strategies. Some of these strategies is to control cameras in a way that the field of view of each camera does not overlap with another view. Other strategies is to control cameras based on a moving object such as a car, person walking, etc... In our system, cameras installed in a surveillance area capture images of a scene based on user's region selection. However, we propose a camera control strategy that covers a region with minimal efforts.

## 2.3 Human Centric Approach

Recently, there has been an emphasis on bringing human in the loop for solving various multimedia analysis tasks [9]. However, surveillance researchers especially from the computer vision community have centered on building automated systems and have hardly adopted a semi-automated approach in which human is involved in the decision making process with the aid of computer generated inputs.

In the following, we describe some of the works that have adopted automated approach. There are some existing automated surveillance systems such as DETEC [7, 33] and Gotcha [14, 33]. Both of these systems are based on motion detectors. When an object appears in the scene, events are triggered and the system stores the triggered events in a central database, such as input images. In our proposed proactive system, when a suspected person appears in the scene, the officer triggers the system about the region of interest, and the system in turn triggers the nearest cameras to cover the specified area.

Another example of surveillance system is the 'Detection of Events for Threat Evaluation and Recognition' which is known as DETER [21]. The purpose of the system is to detect any unusual event of pedestrians and vehicles in outdoor environment and report it. The system depends on two different technologies which are computer vision and threat assessment. Computer vision module is responsible for identifying and tracking

objects and is based on a general approach described in [32], and threat assessment module which is in charge on identifying the threat and classifies it. In this work, computer vision module performs well but it is not clear whether the threat assessment performance could be improved if the number of camera is increased. However, identifying a threat in our system is different. When the officer selects the region, and after identifying the suspected person within captured images, the information about the person will be available to security officer, and based on his observation, he identifies if there is a threat or not.

In [19], a network of cooperative camera for visual surveillance has been presented, where multiple network cameras are engaged to monitor a wide parking area and track multiple moving objects. The system was evaluated to monitor a parking area and a sidewalk. The results showed that the system works correctly on tracking objects, but it is not clear if it performs well when monitoring a crowded area such as in train station or in a shopping mall. However, the cooperation from security officer is needed in our proposed system. The security officer has to monitor the place and identifies the system wherever they see or notice any unusual behavior/activity made people.

Most of the systems that we have discussed above have been developed to be fully automated. In contrast, our proposed system is based on *human-cum-machine* centric approach where human cooperation is needed to assist the surveillance system since we believe that involving the user will improve the overall performance of the surveillance system.

There are a very few works that have adopted a human centric approach for the surveillance purpose. For instance, in [2], a human centric approach has been implemented to assist operator and drive his/her attention to monitor high importance CCTV. The system was evaluated and the human-centric approach presented showed that the

system performs well, and the operator could identify a high percentage of the important events that occurred. In our proposed system, human-centric approach is also presented, where the user play a role where he identifies the region of a suspected person (based on his observation) and also identifies the image and the person within the image from different pictures taken of the region.

Note that, in the proposed system, the quality of the captured image is determined manually by the security officer, which is in contrast to [28]. In [28], a quality-of-view (QOV) measurement has been presented, where the authors defined a QOV metrics to measure the quality of each camera's view to a given subject. The QOV of each camera is helpful for the system to select and determine the best camera installed to perform a video surveillance task on a specific subject. The automatic way of determining quality of an image (or view) suffers from several limitations. For example, such algorithms tend to work in constrained environment. Therefore, we adopt a human centric approach in our proposed system, which allows several pictures to be taken of a suspected person from the cameras installed at different locations. From these pictures, the security officer selects the best image.

Most of surveillance systems are automated systems, where there were not much privileges given to users to utilize the surveillance systems up to full extent. There are a very few works that have adopted a human centric approach for the surveillance purpose, which proved that it had an impact on surveillance efficiency. Therefore, we adopt an intermediary approach in which human and machine works together to maximize the goal of surveillance

# Chapter 3

## Proposed System and Methodology

### 3.1 System Overview

The proposed surveillance system aims to provide the information about a suspected person to the security officer on his/her portable device such as PDA, laptop, etc.

Consider the scenario of a hallway where a security officer watches over the place, walking persons, and there are several cameras that monitor the area. When the officer suspects any person, s/he selects a subregion identifying the current location of that targeted person. In order to do that, a layout view of the hallway is displayed on the officer's portable device which helps him/her to select one or more adjacent subregions. When a subregion is selected, the cameras that are installed in that region are selected to maximize the coverage of the specified area. Thereafter the cameras start sending images of the scene to the security officer for his/her action. Using the application running on the portable device, officer chooses or identifies the suspected person from the received images that are displayed on the application's GUI. Based on the input provided by the officer, the system tracks the specified person in order to capture a good facial image. The face image is then processed for face matching, and the personal information of the suspected

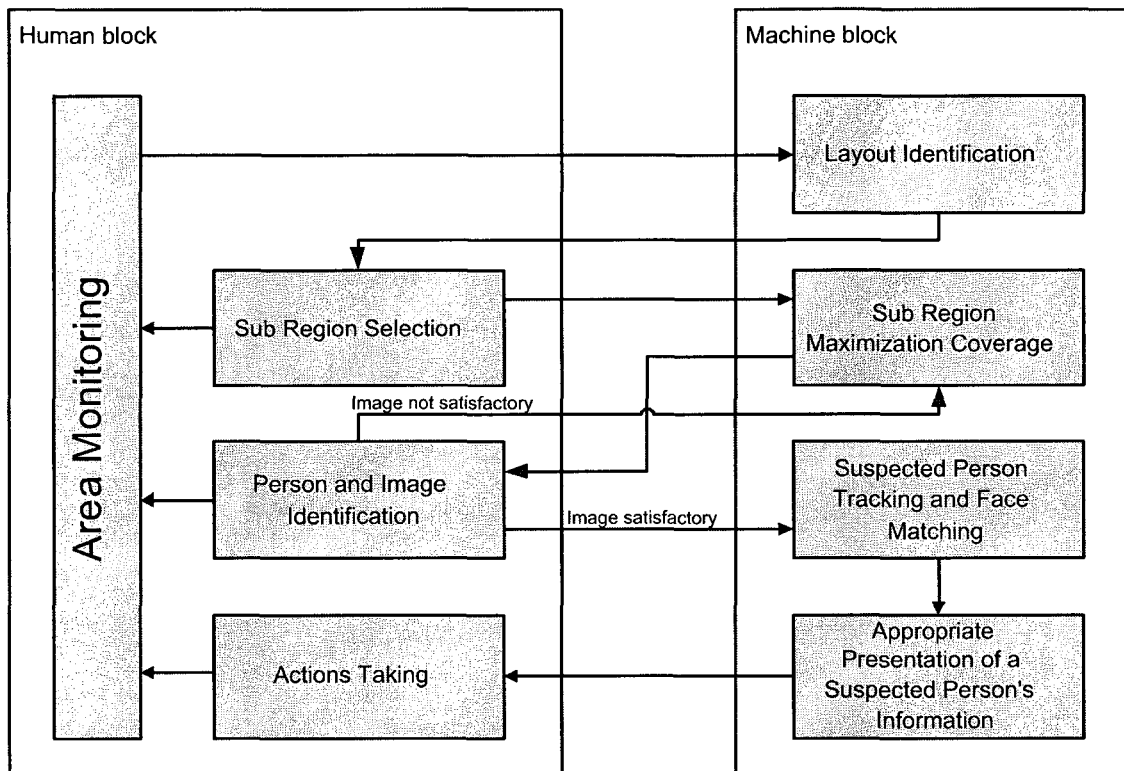


Figure 3.1: Block Diagram

person is extracted and sent to security officer's portable device. Security officer has the option to scan through the information, and based on information provided, the officer takes proper actions, such as add comments to the person's profile, call central station, or arrest the person, etc...

Adhering to the *human-cum-machine* centric approach that we have adopted, the proposed system is divided into eight major components. These components are categorized into two main blocks - 'Human block' and 'Machine block', as illustrated in figure 3.1. The 'Human block' combines all the components which require an action from the security officer, whereas the 'Machine block' consists of the components that respond to the security officer's actions. This chapter briefly describes each of these components.

### 3.1.1 Human block

The components of this block are described as follows:

- **Area Monitoring:** The area monitoring is the process in which the security officer monitors a region of interest e.g. shopping mall, train station, etc... The officer generates an event trigger by using the proposed system's application. The event trigger is generated whenever the officer decides to suspect a person, where this event trigger is then processed by the security system.
- **Subregion Selection:** The layout of the region that is under the supervision of the security officer is divided into subregions and is displayed on the application's GUI. This component helps the officer to easily select the subregion, where the suspected person is located, by dividing the region into small subregions. The officer can select one or more adjacent subregions, and the component is responsible for triggering the system to control the cameras in order to maximize the coverage of the targeted region(s).

*Input:* Subregions of the layout region

*Output:* Subregion selection

- **Person and Image identification:** Several pictures are displayed on the portable device of the security officer, from which s/he chooses one. This component gives the officer the choice to select an image of the person. The inputs provided by officer are sent to an image processing component to process the image and to identify the suspected person in order to retrieve his/her personal information.

*Input:* Images of the scene

*Output:* Picture that contains the suspected person and other related information

- **Actions Taking:** This component allows security officer to take actions based on the retrieved suspected person's information. The actions could be calling the central station for assistance, adding comments to suspected person's profile, arresting the person, etc...

*Input:* Suspected person's information

*Output:* Action(s) taken by officer

### 3.1.2 Machine block

The components of 'Machine block' are described as follows:

- **Layout Identification:** is responsible for identifying the layout region that is under security officer's supervision. Layout identification is initiated based on an event trigger such as, a command entered by officer for layout view, in which the geographic position of the security officer is identified by a GPS system. The component is responsible to identify the layout region and a list of cameras that are installed in that region.

*Input:* Event trigger (e.g. command by the officer)

*Output:* List of cameras and layout region that is under the supervision of the officer

- **Subregion Coverage Maximization:** When the subregion is selected via application's GUI, the system triggers the cameras that cover the area to maximize the coverage. The corresponding cameras starts panning/tilting to the targeted region and starts sending pictures of the scene.

*Input:* Subregion selected

*Output:* Pictures taken by each of the cameras that cover the area

- **Suspected Person Tracking and Face Matching:** Once the picture and person are identified within a picture, the component is responsible for tracking the targeted person to get a good face image. When the facial image is available the component is then responsible for extracting the face features of the person and matching it.

*Input:* Picture containing suspected person

*Output:* Suspected person's personal and contact information

- **Appropriate Presentation of a Suspected Person's Information:** is responsible for displaying available information of the suspected person to the officer. Personal and other background information are displayed and available for security officer to take any actions needed.

*Input:* Suspected person's information

*Output:* Suspected person's information presentation

In the following subsections, we present the three main constituent of our contribution in this thesis. These are system architecture, *human-cum-machine* centric approach, and camera control strategy for coverage maximization.

## 3.2 Proposed Architecture

We introduce in our proposed system the concept of Web Services technologies. Unlike the traditional client/server approach, web services share business logic, processes, and data across the network [10, 37]. Since the technology is based on open standards such as HTTP, XML-based protocols including SOAP [35] and WSDL [36], where the communication over the network is in XML, web services are not tied to a specific programming language, or operating system. Web services technologies allow developer to

develop services and deploy them to a server so they can be called and used in a secure and reliable way. Another advantage of using web services is that it allows integrating or replacing any existing modules to an application instead of redeveloping them. For instance, ‘VeriLook library for face detection and identification’ [20] used in our proposed system, could be later replaced by any service in case that service provides better results without obstructing any existing code.

Driven by the web service technology, the proposed system architecture is divided into the following three main components: Multimedia Web-Services, Intelligent Decision Web-Services, and Application’s Graphical User Interface. The architecture is shown in figure 3.2. In the following, we will describe the sub-components of each of the three major components.

### 3.2.1 Multimedia web-services

This component is composed of various multimedia web-services. In particular, we have used four web services, which are described as follows:

- **Sensory Streams:** Sensory streams provides web service interface of a sensor installed, such as camera, in a surveilled area. Other components could communicate with this web service in order to control the sensor, or provide streaming information about that sensor.
- **Sensory Environment Metadata Repository:** Installed sensors in a surveilled area are subject to geographic installation’s changes or failure. The ‘Sensory Environment Metadata Repository’ web service provides functionalities such as updating the location of a sensor, disabling/enabling an installed sensor for monitoring, adding/deleting a sensor, etc. The component also provides the functionality to retrieve information about a specific installed sensor such as its initial location

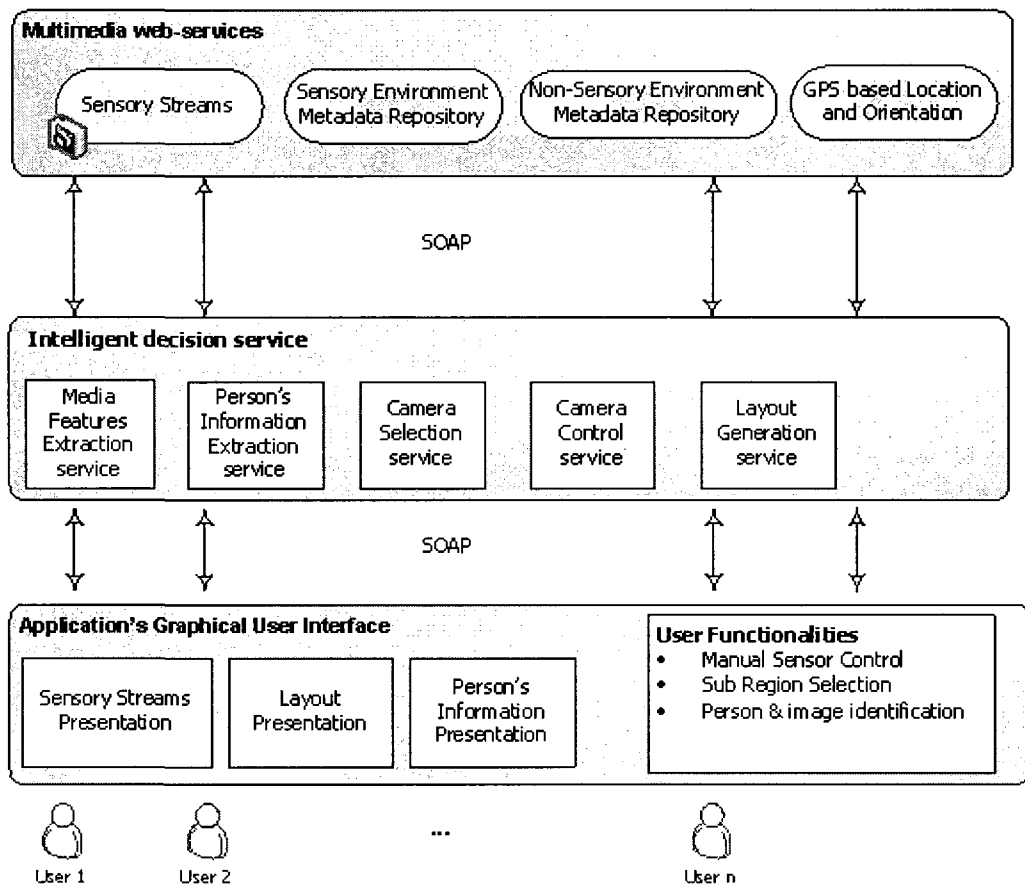


Figure 3.2: High Level System Architecture

within the surveilled area, current orientation, IP address, etc. The information of each sensor is stored in a database, and when the component receives a SOAP request, it queries the database based on each request and responses back with the requested information. The component could be accessed using a web application that to be provided to the administrator, and could also be invoked by other web services components.

- **Non-sensory Environment Metadata Repository:** ‘Non-sensory Environment Metadata Repository’ web service is responsible for providing information about landmarks installed in a surveilled region, such as ATM machine, telephone booth, etc. Similar to installed sensors in a surveilled region, landmarks are subject to geographic installation change where some of the landmarks could be removed from their current location or placed to another location. In addition to that, new landmarks could be introduced to the area, where this component provides the functionality to add new landmarks, delete existing landmarks or change/update information of an existing landmark.

Figure 3.3 shows the interaction between client application, ‘Sensory Environment Metadata Repository’ and ‘Non-sensory Environment Metadata Repository’ web services components, and the database when adding a sensor and a landmark to the system.

- **GPS based Location and Orientation:** ‘GPS based Location and Orientation’ provides in real time the current location of the security officer and the direction s/he is looking to. Knowing the current location of the security officer, the system could identify the surrounding area of the security officer. For instance, it could determine the landmarks and sensors installed to be displayed on the client’s graphical user interface. When a SOAP request is invoked by the client, the component

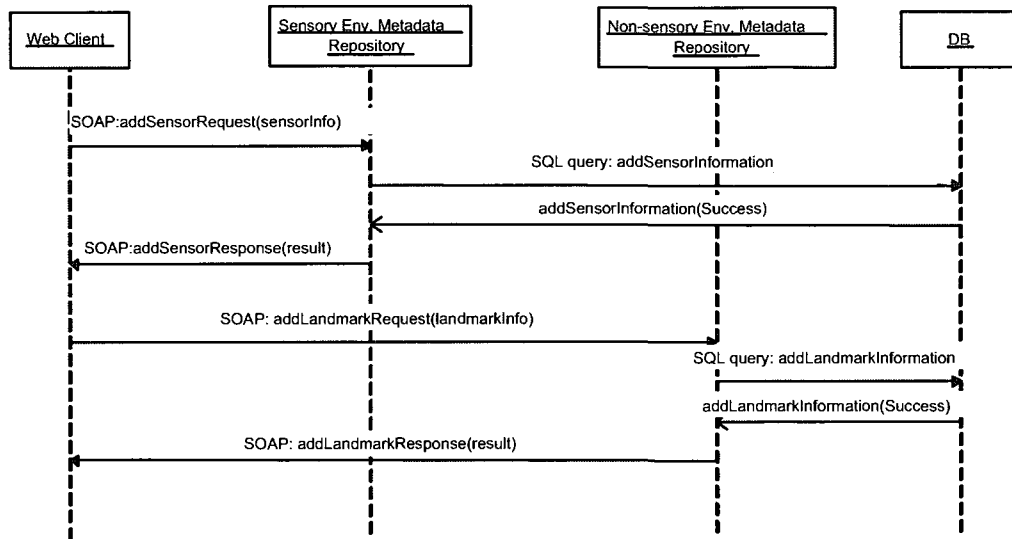


Figure 3.3: Adding sensor and landmark to the system

accesses the GPS system (through API the GPS provides) hold by the security officer, in order to provide his/her current location. Once the information is retrieved, the SOAP response is sent back to the requestor.

### 3.2.2 Intelligent decision service

This set of web services provide business logics and use intelligent algorithms to respond to the users' requests. The web services can also communicate internally with each others, beside their external communications with the other two main components of the architecture.

Various web services used in this component are described as follows:

- **Media Feature Extraction and Comparison:** The web service component is responsible for extracting and comparing face features from a media stream such an image. This component uses VeriLook Face Identification SDK [20] where it enables the component to perform face feature extraction and comparison with other face

features. ‘Person’s Information Extraction’ service which will be described later, uses the face image extraction and comparison features provided by the VeriLook SDK in order to extract the face feature from an image, and compare it with face features stored in the database to retrieve the personal information of the captured person.

- **Person’s Information Extraction:** ‘Person’s Information Extraction’ service is responsible for extracting personal information of a person captured in an image. We supposedly have information about users in a database such as personal, contact information, and face features of each user. When a picture of a person is available for comparison, the web service communicates with the ‘Media Feature Extraction and Comparison’ service in order to extract the face feature and compares it with the ones stored in the database. When there is match between the face feature extracted and a face feature stored in the database, the corresponding information of the person is queried from the database and sent back to the requestor.

Figure 3.4 shows the interaction between components when a request is sent to ‘Person’s Information Extraction’ service component to retrieve personal and contact information of a person.

- **Camera Control:** ‘Camera Control’ service provides a method of controlling a selected cameras in order to maximize the coverage of a region specified by the officer. The service implements our proposed camera control strategy introduced later in this chapter. This service communicates with ‘Sensory Streams’ component to control a specific camera.
- **Camera Selection:** ‘Camera Selection’ service provides a mechanism to select sensors for a particular action. The service interacts with the ‘Sensory Environ-

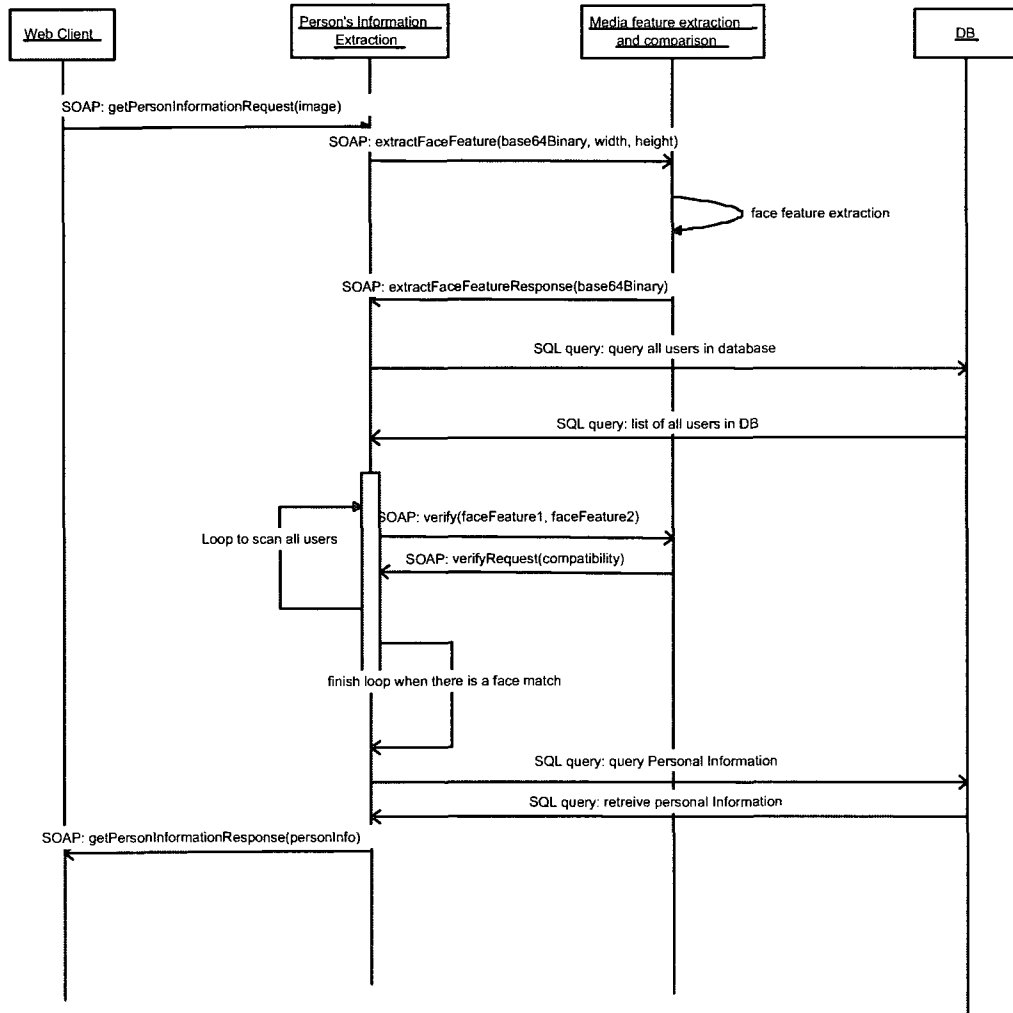


Figure 3.4: Retrieving personal information

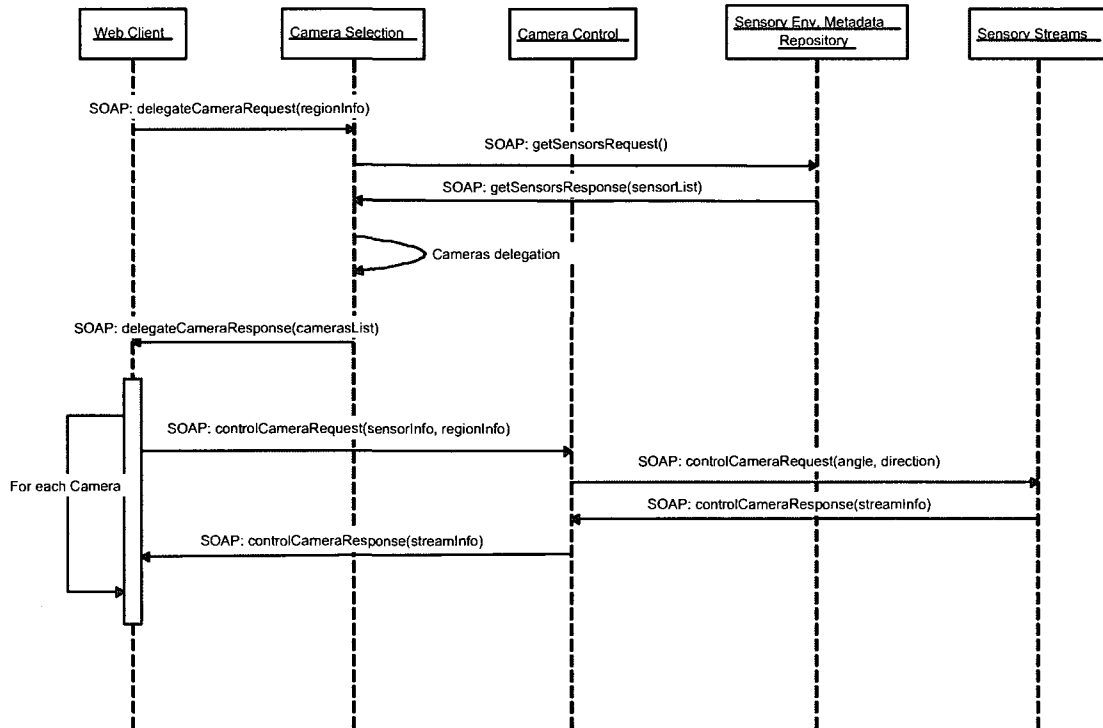


Figure 3.5: Delegating and controlling list of cameras

ment Metadata Repository' service to learn about the cameras installed in a region and subsequently determines or delegates cameras to cover a specific subregion. The component is usually invoked by the client's graphical user interface in order to determine the cameras delegated to cover a subregion specified. The camera selection algorithm delegates a number of cameras to cover the subregion and sends back the SOAP response with the information about each delegated camera to the requestor.

Figure 3.5 shows the interaction between client, 'Camera Control' and 'Camera Selection' services of a scenario covering a subregion.

- **Layout Generation:** This service is used to generate the layout to be displayed

on officer's portable device. The layout display takes into consideration the current location of the officer and his/her path direction. This service communicates with the 'Sensory Environment Metadata Repository' and 'Non-sensory Environment Metadata Repository' services to learn about the installed sensors and landmarks, hence to determine the sensors and landmarks to be displayed based on the current location of the officer provided by the 'GPS based Location and Orientation' service.

The implementation of each of web service is independent of other web services. The multimedia web services interface provides an interface to retrieve or update information about any sensor or landmark installed in the region. The intelligent web services components provide complex functionalities, and substituting any of the components by another, do not affect the functionality of the overall system as long as the new component provides the same interface as the substituted one. For instance, if a new web service introduces a better face feature and comparison algorithm, the 'Media Feature Extraction and Comparison' web service could be replaced or substituted without affecting the overall functionality of the system, but it could improve the effectiveness of the system's output result.

### 3.2.3 Application's Graphical User Interface

- **Layout Presentation:** This component is responsible for displaying the layout of the region where the security officer is located, as well as the landmarks and sensors located in that region. The component invokes the 'Layout Generation' service in order to attain the layout information. (Figure 3.6)
- **Sensory Streams Presentation:** The 'Sensory Stream Presentation' is responsible for providing sensory streams from installed sensors. At initial state, the system

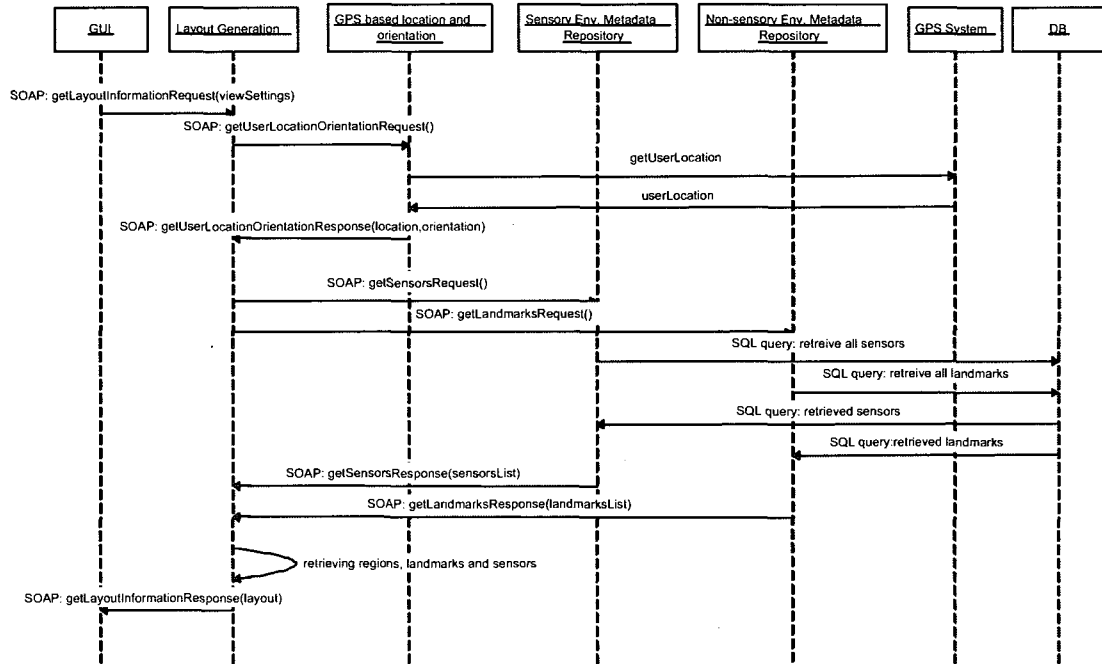


Figure 3.6: Layout generation and presentation

application communicates with the ‘Camera Selection’ in order to determine the nearest sensors or cameras to the security officer. Once the list is available, the application communicates with each camera’s hardware in order to provide streams in real time.

- Subregion Selection:** A region will be divided into several subregions (will be described later in details in chapter 4). The user has the option to select a subregion for coverage maximization. When a subregion is selected by the officer, the application sends a SOAP request to ‘Sensor Selection’ service which provides a list of cameras that are delegated to cover the subregion. When the list of cameras is retrieved, the application sends a soap request to ‘Camera Control’ service to control each cameras delegated by the selection component.
- Voice Speech Enabled:** Security officer has multiple tasks to do when watching

over a place. However, in order to ease the officer from other things while watching over the place, the application is built to be a voice speech-enabled, where officers are able to execute basic commands verbally.

### **3.3 *Human-cum-machine* Centric Approach**

We advocate for a *human-cum-machine* centric approach for building the proposed surveillance system. As discussed in chapter 2, most of surveillance systems are automated where there are a very few works that have adopted a human centric approach, which proved that it had an impact on surveillance efficiency. Therefore, an intermediary approach is adopted in which human and machine works together to maximize the goal of surveillance.

In this section, we discuss the role that the user plays towards our proposed proactive system, and we discuss in details the application's graphical user interface in chapter 4.

#### **3.3.1 User's Roles**

The system is accessed by two different categories of users: 'Security Officer' and 'Administrator'. Each of these categories has different privileges.

The privileges given to the security officer are basic privileges comparing to the ones provided to the administrators. Security officers are able to view and personalize the layout view, select a subregion, identify suspected person from the pictures provided by the cameras, etc. However the administrators who are monitoring the system have access to more advanced features which are related to the environment setup such as adding, removing or enabling a landmark/camera to the system. Figure 3.7 shows the UML use case diagram, which presents a graphical overview of the functionalities provided by the



Figure 3.7: UML Use Case Diagram

system and the goals of each category of users.

### 3.4 Camera Control

Several cameras are installed to monitor a hallway. The general problem is to maximize the coverage of a selected region by the security officer. We assume that the layout of the surveilled region is convex where the region is defined by:

$R_i$  : Subregion  $R$  of the convex region, where  $1 \leq i \leq r$

Each of the regions  $R$  is defined by:

$s$  : Number of sides that region  $R_i$  consists

$L_z$  : Length of each side of a region where  $1 \leq z \leq s$

There are  $n$  PTZ cameras where  $C_1 \dots C_n$  installed in a surveilled area, and each of cameras  $C$  is defined by:

$x_c$  : Position  $x$  of the camera in  $x, y - Plane$

$y_c$  : Position  $y$  of the camera in  $x, y - Plane$

$\alpha_r$  : Maximum orientation that the right side line could achieve

$\alpha_l$  : Maximum orientation that the left side line could achieve

$i\alpha_r$  : Initial right angle

$i\alpha_l$  : Initial left angle

$\beta$  : View angle of the camera

The objective is to determine the new orientation of the cameras in order to maximize the coverage of a given surveilled area.

### 3.4.1 Determining possible angles to cover a subregion in 2-D

To determine the orientation of each of selected cameras, we will only study the case of one camera, which therefore could be applied for all other moving cameras. Since each of the cameras might have different parameters than the others such as the angle view, orientation, we assume that each of these parameters is defined, as well as the positions and orientation in the hallway. Suppose a camera  $C$  installed in the hallway. The position of the camera in the hallway is known for the camera since the camera is fixed and is not moving. We denote the point  $C$  in  $x, y - Plane$  as the camera object which has the coordinate  $C(x_c, y_c)$ . For simplification, the point  $C$  of the camera is chosen as the origin of  $x, y - Plane$ , which means the camera has the coordinate of  $C(x_c = 0, y_c = 0)$ . The objective of the camera is to cover the area selected by a security officer. The hallway is divided into several small convex subregions, and each of the subregions is a type of a circle. We represent the point  $O$  in  $x, y - Plane$  as the center of the subregion selected; where the center of the region in  $x, y - Plane$  has the coordinates of  $(x_o, y_o)$  and radius  $r$  (See Figure 3.8). Since the coordinates of the circle is known, we can represent the equation of the circle which is:

$$(x - x_o)^2 + (y - y_o)^2 = r^2 \quad (3.1)$$

The initial known values of the camera are  $i\alpha_r$  and  $i\alpha_l$  in respect to the x-axis of the  $x, y - Plane$ . These values represent the initial orientation of the camera.

As shown in figure 3.8, the camera's view plane consists of two lines that intersect at point  $C$ . We represent the two lines as the right side line and the left line side which have the following equations.

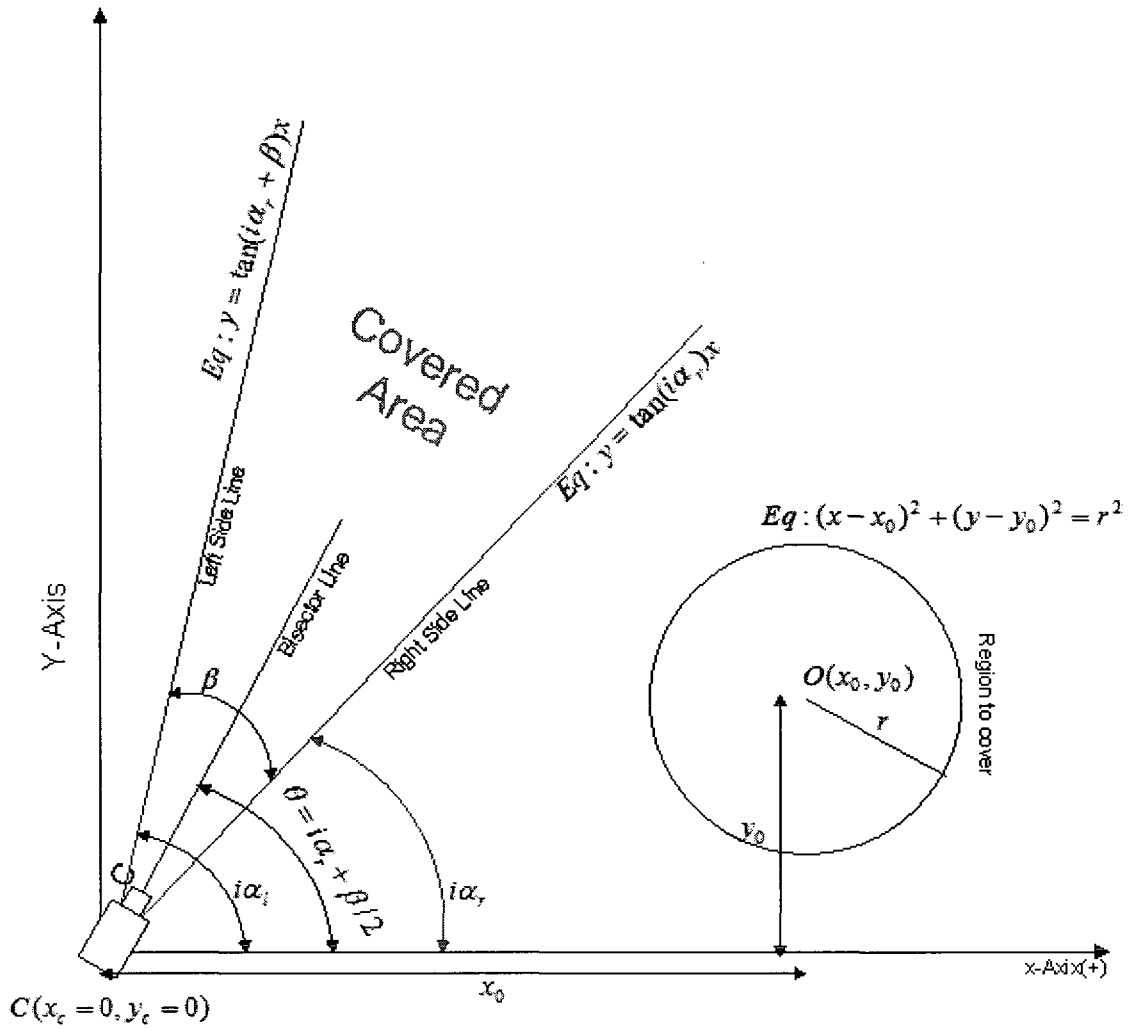


Figure 3.8: 2D plan view showing initial camera's state

**Right side Line:**

$$y = \tan(i\alpha_r)x \quad (3.2)$$

**Left side Line:**

$$y = \tan(i\alpha_r + \beta)x \quad (3.3)$$

We also define the bisector line which divides the angle view into two equal angles; each angle has the value of  $\beta$ .

**Bisector Line:**

$$y = \tan(\theta)x \quad (3.4)$$

where

$$\theta = i\alpha_r + \beta/2$$

We need to determine the new orientation of the camera; therefore we need to determine the angle the camera should pan in order to maximize the coverage. To minimize the panning time of the camera and maximize the coverage, the right or left side line should barely touch the circle at a single point, in other words one of the side line should be tangential to the circle. Since point  $C$  is external to the circle region, there exist two lines that touch the circle at two different points and are tangential to the circle. The two points are denoted by  $M$  and  $N$  which we will determine in this section (See figure 3.9). By definition, a circle  $O'$  with a diameter  $CO$ , intersects the circle 3.1 into two points that form with point  $C$  two tangential lines to the circle region. The center of the circle  $O'$  has the coordinate  $(x_o/2, y_o/2)$  and a radius  $R$  which is defined by:

$$R = \sqrt{\frac{x_o^2 + y_o^2}{4}} \quad (3.5)$$

Knowing the radius  $R$  defined in equation 3.5 and the coordinate of the center of the circle  $O$  defined in 3.1, we can define the equation of the new circle  $O'$  which is defined by:

$$(x - x_o^2) - (y - y_o^2) = \frac{x_o^2 + y_o^2}{4} \quad (3.6)$$

An intersection between two circles defines a radical line that is perpendicular to the line formed by the center of the two circles. Therefore, intersection of circle  $O$  determined in 3.1 and circle  $O'$  determined in 3.6, the radical line is defined which has the following equation:

$$(x - x_o^2) - (y - y_o^2) = \frac{x_o^2 + y_o^2}{4} \quad (3.7)$$

The radical line shown in figure 3.9 intersects the circle  $O$  and  $O'$  into two points. Each of the points forms with the camera  $C$  a tangent line to the circle  $O$ . Putting equation 3.7 and equation 3.1 together, the two points  $M$  and  $N$  are defined which have the following coordinates respectively:

$$M\left(\frac{Cx_o + C^{1/2}y_or}{x_o^2 + y_o^2}, \frac{Cy_o - C^{1/2}x_or}{x_o^2 + y_o^2}\right) \quad (3.8)$$

$$N\left(\frac{Cx_o - C^{1/2}y_or}{x_o^2 + y_o^2}, \frac{Cy_o + C^{1/2}x_or}{x_o^2 + y_o^2}\right) \quad (3.9)$$

where

$$C = x_o^2 + y_o^2 - r^2$$

The angle formed by points  $C$  and  $M$  defined in 3.8, as well as angle formed by points

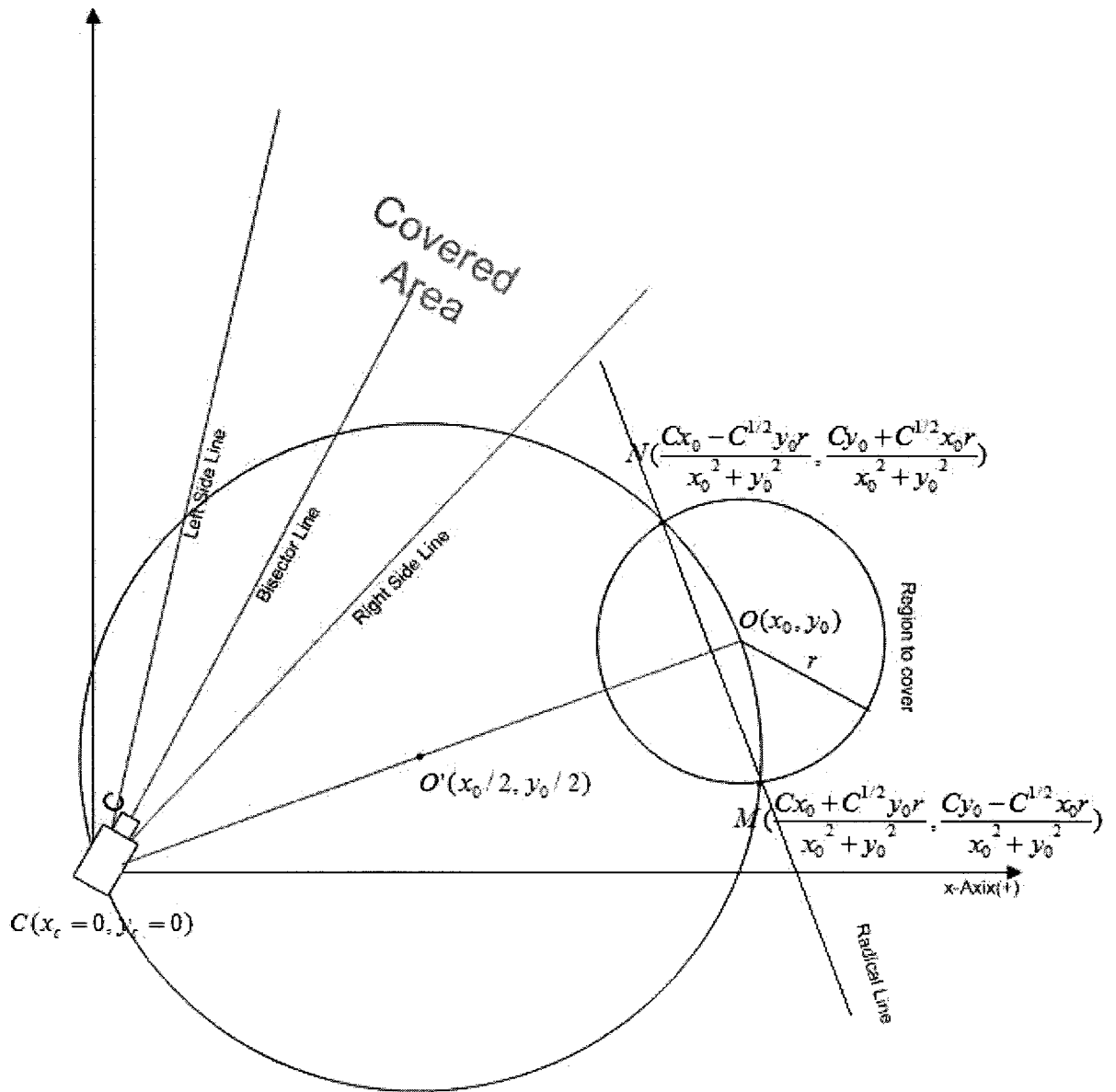


Figure 3.9: 2D plan view showing the region to cover

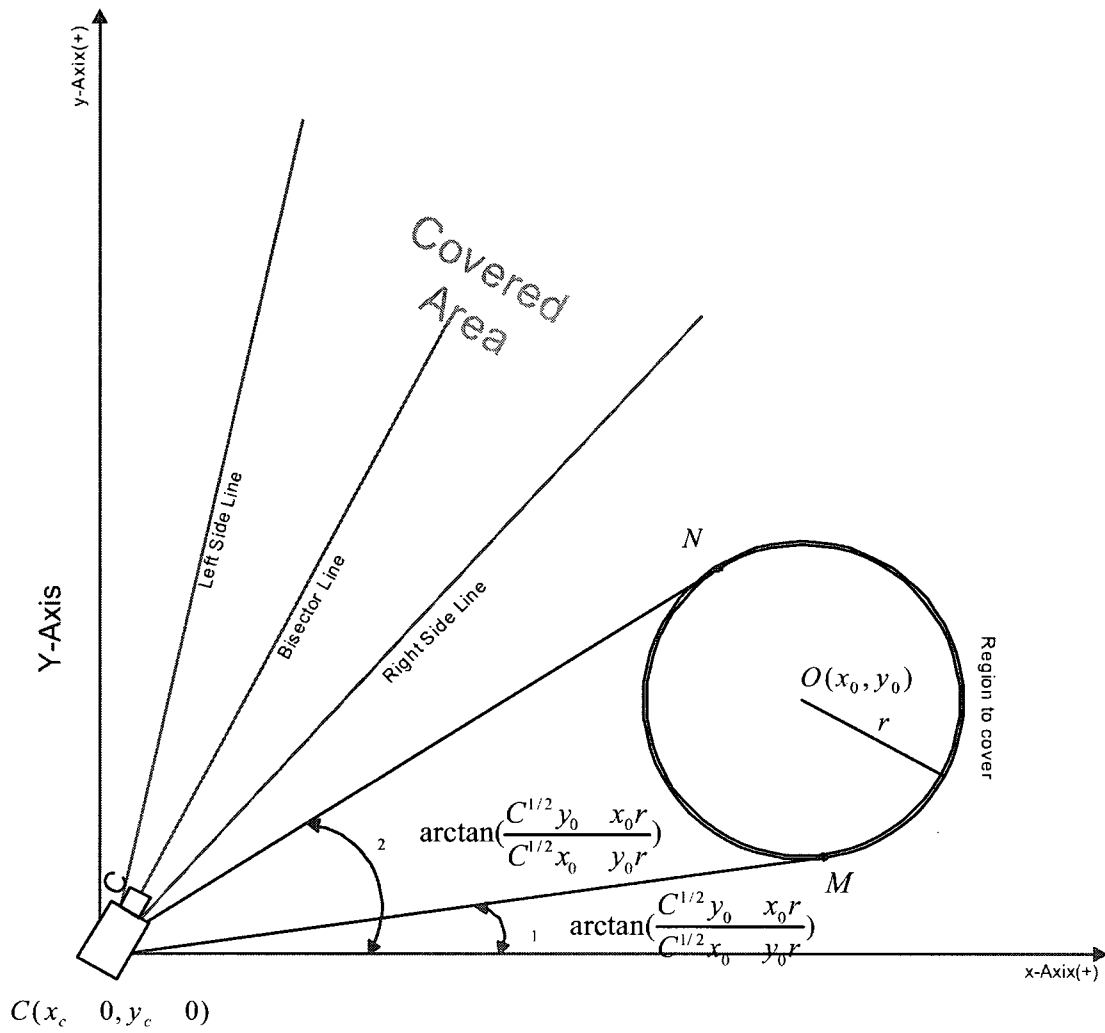


Figure 3.10: 2D plan view showing the new panning angles to cover the specified region

$C$  and  $N$  in 3.9 with the x-axis, shown in figure 3.10 are defined respectively as follows:

$$\phi_1 = \arctan\left(\frac{C^{1/2}y_o - x_or}{C^{1/2}x_o + y_or}\right) \quad (3.10)$$

$$\phi_2 = \arctan\left(\frac{C^{1/2}y_o + x_or}{C^{1/2}x_o - y_or}\right) \quad (3.11)$$

### 3.4.2 Determining a camera's panning angle, and panning direction

Let us denote by  $C$  an installed PTZ camera in a hallway and  $AllCam$  the set of all  $n$  installed PTZ cameras, where

$$AllCam = C_1, C_2 \dots C_n \quad (3.12)$$

Our simple strategy delegates the nearest cameras that belongs to 3.12 to take pictures of a scene. Let us denote by  $SelCam$  the set of selected PTZ cameras where  $SelCam \subset AllCam$ , and

$$SelCam = \forall s \in AllCam / C_s < d \quad (3.13)$$

where  $d$  is the distance between the center of a subregion and PTZ camera  $C_s$ .

As shown in the previous section, two angles  $\phi_1$  (3.10) and  $\phi_2$  (3.11) were determined. Each of the selected cameras in  $SelCam$ , could pan to the right or to the left to cover a selected subregion. Since two angles were determined in 3.10 and 3.11, each camera has four possible ways to cover a subregion, where a PTZ camera could pan towards clockwise or counter clockwise. In our strategy, we want to cover a subregion with minimal amount

effort made by a camera.

Each of the installed camera has a maximum angle to pan right and maximum angle to pan left based on its current orientation. We denote by  $max_R$  the maximum angle that the camera could pan right, and  $max_L$  the maximum angle that the camera could pan left.

The maximum effort the camera could make to the left and to the right are determined based on the camera's current orientation.

Since there are four possible ways, we could determine the four possible angles  $angle_1$  ...  $angle_4$ .

$$angle_1 = i\alpha_r - \phi_1 \quad (3.14)$$

$$angle_2 = \begin{cases} angle_1 - 360 & \text{if } angle_1 > 0; \\ angle_1 + 360 & \text{if } angle_1 \leq 0. \end{cases}$$

$$angle_3 = i\alpha_l - \phi_2 \quad (3.15)$$

$$angle_4 = \begin{cases} angle_3 - 360 & \text{if } angle_3 > 0; \\ angle_3 + 360 & \text{if } angle_3 \leq 0. \end{cases}$$

Four possible angles are determined  $angle_1$  ...  $angle_4$ . Our camera control algorithm delegates one these possible angle to send to the corresponding camera for control. Note that, the sign of an angle only identifies the panning direction of the camera, for example, -30 degrees means that the camera is to pan 30 degrees counter clockwise. The algorithm used for angle delegation is described as follows:

- *Step 1:* Place all four possible angles  $angle_1$  ...  $angle_4$  in a queue

- *Step 2:* Pick the smallest angle in value from the queue, if queue is empty go to step 7, otherwise remove the angle from the queue, assign it to  $\alpha$ , and proceed to step 3.
- *Step 3:* Check the sign of the angle  $\alpha$ , if it is either negative or positive
- *Step 4:* If the sign is negative, the camera would pan counter clockwise if the specified angle has been delegated. Therefore,  $\alpha$  has to be checked and compared with the  $max_L$  if the sign is negative or with  $max_R$  if the sign is positive.

The following rules apply for angle  $\alpha$  to be delegated:

When sign is negative:

*if*( $|\alpha| \leq max_L$ ) proceed to step 6, otherwise go to step 2

When sign is positive:

*if*( $|\alpha| \leq max_R$ ) proceed to step 6, otherwise go to step 2

- *Step 5:* The camera is not able to cover the subregion
- *Step 6:* The angle  $\alpha$  is delegated and send to the corresponding camera for control
- *Step 7:* The queue is empty, the camera could not cover the region

Finally, each camera has four possible angles which are defined in 3.14 and 3.15 to cover a subregion which. Following our simple strategy that previously discussed, one of these possible angles is delegated to be applied to the camera for control.

# Chapter 4

## Proposed GUI, and System Implementation

In this chapter, the application graphical user interface is discussed, as well as the system implementation.

### 4.1 Application's graphical user interface

The graphical user interface is an important aspect to be considered in the system implementation. The overall system acts based on user's input or trigger such as the selection of a region, and therefore affects the overall effectiveness of the system.

Our designed application's graphical user interface is consisted into four components [3]: Camera View, Map View, Personal Info View, and Scene Picture View. Each of these components is described as follows.

### **4.1.1 Camera View UI Component**

The “Camera View UI” component provides different scene views from multiple installed cameras. Based on the location of the user in the hallway, the nearest cameras to the security officer are selected and delegated for scene display. The “Camera View UI” gives the user the ability to control one of the multiple cameras that are delegated, by requesting to pan right/left or tilt up/down. The user has also the ability to switch between cameras’ views, by selecting and activating one of the other cameras for a manual control. The user is capable to manually control any camera installed with the help of the “Map View” component which will be discussed later in this chapter. The following is a summary of the functionalities that the “Camera View UI” component provides. The user has the ability to identify the number of the scene views to be displayed to “Camera View UI”.

#### **Functionalities**

- View multiple camera views
- Identify the number of scene views to be displayed
- Manually Control one of the cameras displayed for view
  - Pan left, right
  - Tilt up, down
  - Zoom
- Switch between camera views for manual control

### **4.1.2 Map View UI Component**

“Map View UI” component is where the security officer selects a subregion of the hallway or selects a camera for manual control. The “Map View UI” component displays the current map which indicates the position of the security officer as well as his/her surroundings. The region displayed where the security officer is residing, gets divided into small subregions for selection. The division of the subregions are based on the settings previously set by the officer. The officer could identify the size of a subregion, identify the number of subregions to be shown vertically and horizontally on the layout view, and identify the display option. Two approaches have been proposed for the display option and presented later in this section. In addition to that, the component indicates officer’s current location as well as the cameras installed in hallway. To help the user easily identify a region, an option has been introduced which allows the user to view landmarks such as an ATM machine, telephone booth, etc. But due to the limited size of portable devices, we decided to implement the system with the options of enabling or disabling one or more landmarks, so that the interface does not become crowded and therefore distracts the user. This approach will be analyzed and evaluated in term of user’s distraction and user response time.

Finally, the user has the ability to manually control a camera by selecting the desired camera from the map view display for manual control.

#### **Functionalities**

- Select subregion to maximize its coverage
- Display/hide landmarks or cameras installed
- Select camera for a manual control from “Camera View UI” component

- Maximizing/minimizing the dimension of a subregions
- Defining the number of subregions to be displayed on the screen

The regions divided are based on the settings entered by the user, which specifies the number of the subregion to be drawn and the size of a subregion. Dividing the regions into small subregions without taking into account user's location, could result the user to be placed in the middle, or near the edge of a subregion. For that reason, we have decided to implement another approach which would place the officer at the corner of four subregions. We have named the two approach as Dynamic Layout display Approach (DLDA), and Fixed Layout display Approach (FLDA). The difference between the two approaches is that one of the approaches takes into consideration the location of the security officer, and the other approach takes into consideration the hallway shape and dimension in displaying the subregions.

#### **Approach 1: Dynamic Layout display Approach (DLDA)**

In this approach, the subregions are drawn dynamically based on the location and direction of the security officer. Suppose an officer is standing in the mid of the hallway as shown in figure 4.1. The 'Star' represents the security officer's location, and the arrow indicates the direction or the path the officer is taking. In Dynamic layout approach, the subregions boundaries are dynamically generated based on current location and officer's direction. In other words, the security officer will be located at the boundaries of four subregions as shown in figure 4.2 where the squares represent the subregions that are dynamically drawn. The "Map View UI" component will display the map in figure 4.2 but rotated as shown in figure 4.3. The number of regions to be drawn and the dimension of each region could be set by security officer using the settings option.

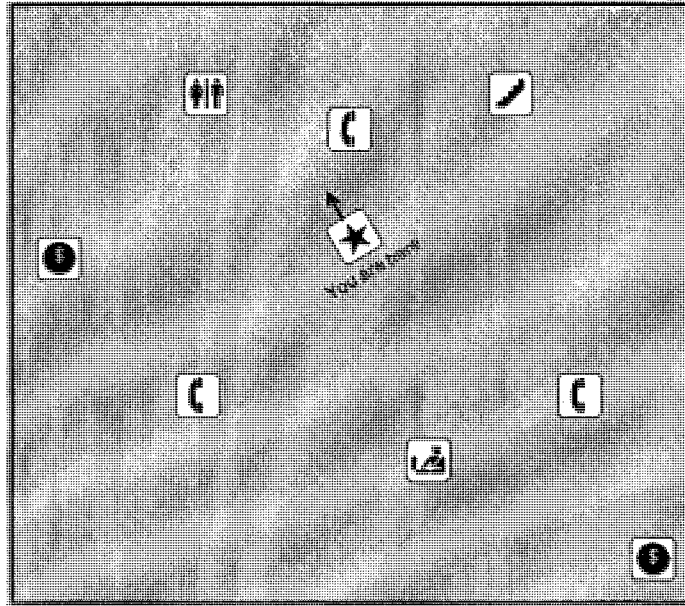


Figure 4.1: Officer in the hallway

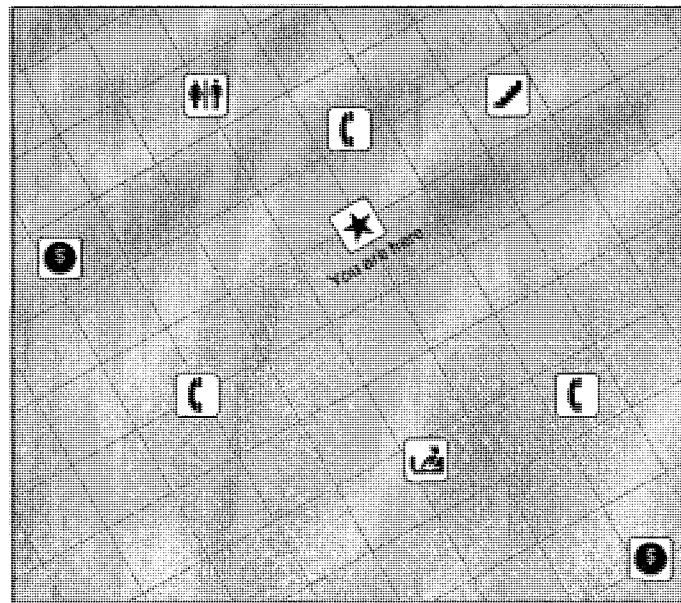


Figure 4.2: Dynamic Layout display: Subregions representation

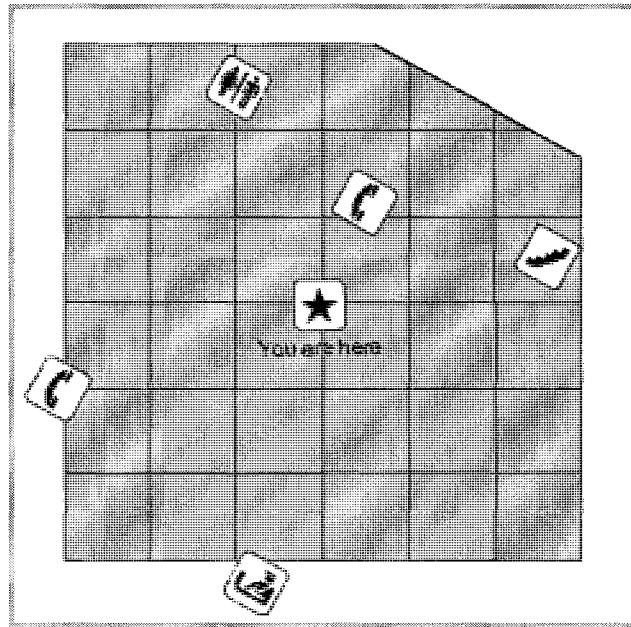


Figure 4.3: Dynamic Layout display: Rotated map to be displayed on the GUI

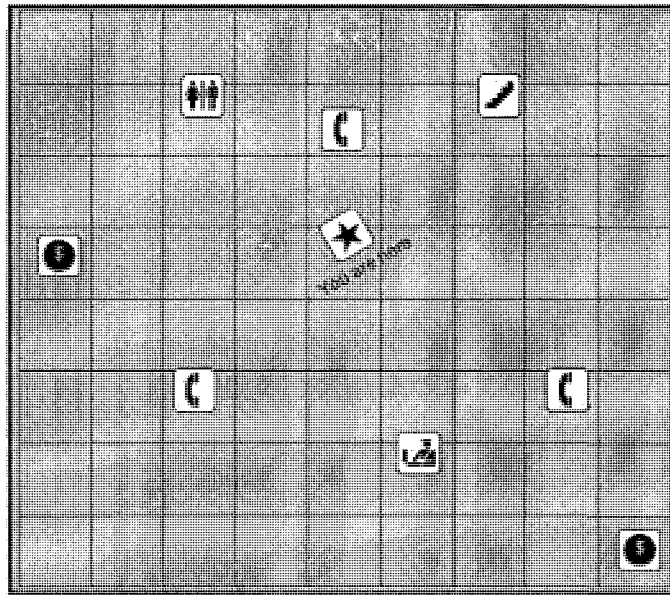


Figure 4.4: Fixed Layout display: Subregions representation

### Approach 2: Fixed Layout display Approach (FLDA)

The difference between the fixed and dynamic layout approach, is that the fixed layout approach does not take into consideration the current location of the security officer and the path is taking. The subregions are generated based on the hallway shape and size as shown in figure 4.4. However the same technique presented in DLDA is used for displaying the image on the 'Map View UI'. The user has also the ability to change the number of regions and their size anytime using the settings option. Figure 4.5 shows the map that will displayed on the 'Map View UI' component.

#### 4.1.3 Personal Info View Component

The personal information view provides the officer with some information about a suspected person. The information is grouped into three categories: Personal Information category provides personal information such as name, birth date, and profession, Con-

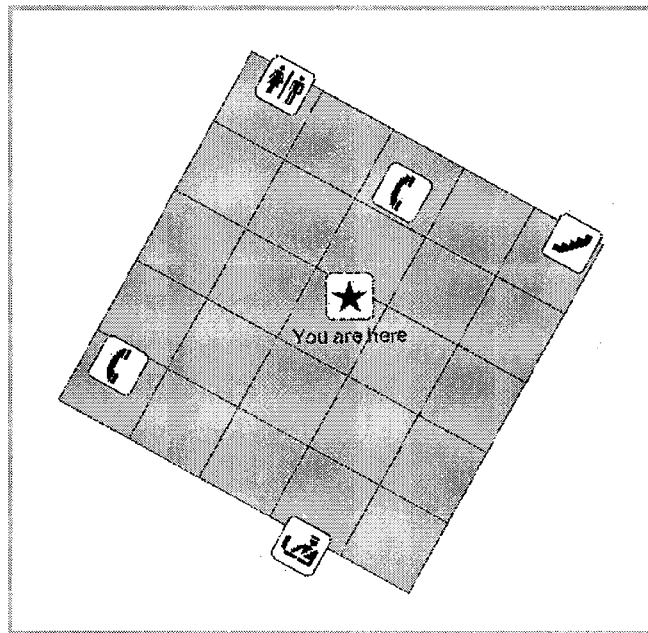


Figure 4.5: Fixed Layout display: Rotated map to be displayed on the GUI

tact Information category provides the contact information of the suspected person, and Activity category offers an overview of recent activities that have been entered and submitted by previous security officers. The user has the option to add activity comment for the suspected person, so that it can be seen by other officers if they suspect the same person in the future.

**Functionalities**

- Scan/View Personal, Contact and Activity information
- Add activity comment

**4.1.4 Scene picture UI Component**

The view displays the picture of the scene taken from several cameras installed. The officer has to identify the image and the person suspected from the images displayed. If the user could not identify the suspected person from the current scene pictures, the component refreshes its pictures view by displaying new scene pictures.

**Functionalities**

- User is able to identify image and person within the image
- New pictures of the scene are displayed in case the previous images are not satisfactory

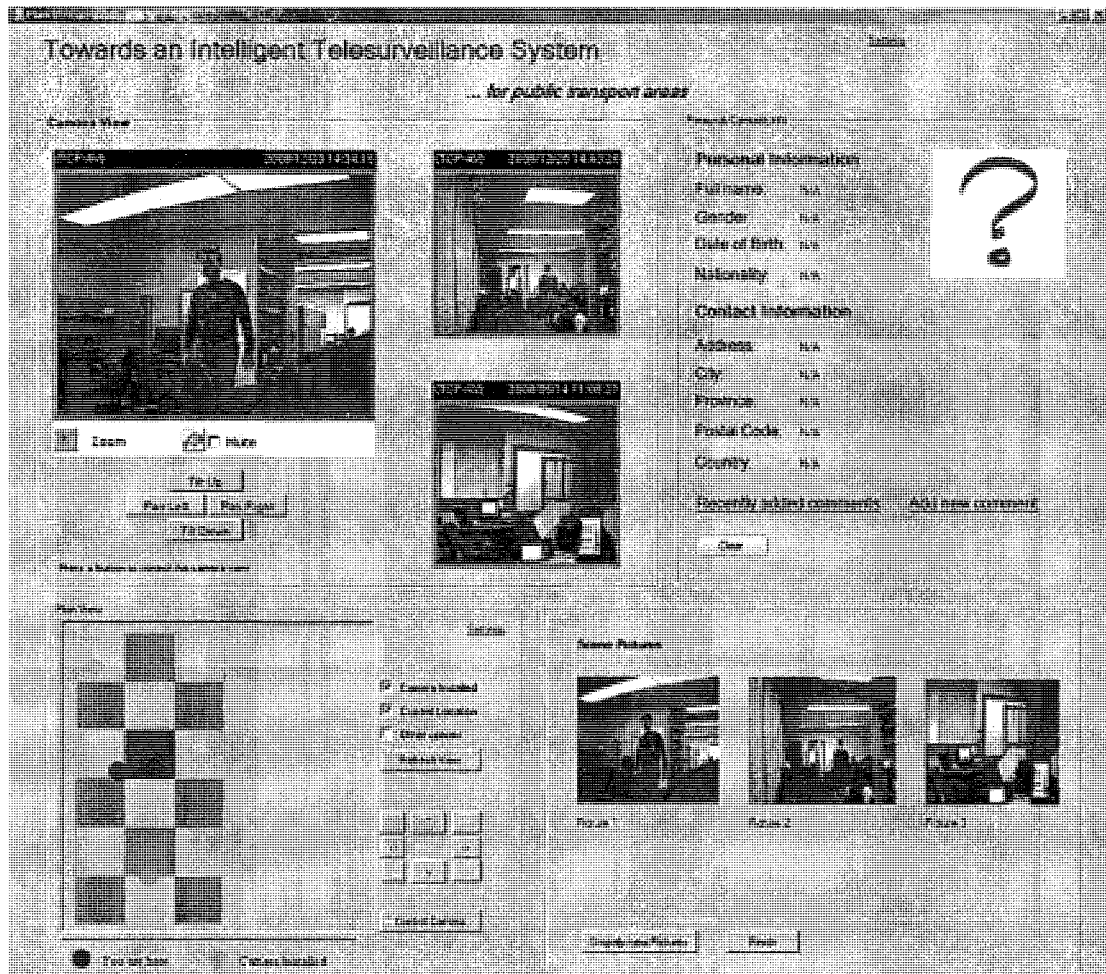


Figure 4.6: Screenshot of the system application

## 4.2 System implementation overview

In this section, we will list and describe the main technologies used in the security system. The section is divided into two main parts: Client Application and Web Services Components.

### **4.2.1 Client Application**

The client application utilizes multiple technologies. One of the main technologies that the application uses is C# which is a part of .NET framework. The technology facilitates the creation of application's windows form, where it provides the developer the ability to create or reuse user interface (UI) controls, and thus reduce the amount of the code by encapsulation the common functionality in the UI controls. The "Map View" component of the client application requires drawing a map of the hallway, where System.Drawing which is also a part of .NET framework is used.

#### **Microsoft Speech Recognition SDK**

The voice speech-enabled application developed uses the software development tool which is the Microsoft Speech SDK [6]. The development kit is for building speech engines and includes a collection of speech components for managing audio, events, and grammar compiler. The application accepts user's input and matches the input against the grammar to recognize the word or command spoken by the user. When words or commands are recognized by the speech engine, the application receives events and performs actions based on events received.

#### **Architectural Pattern**

The Model-View-Controller (MVC) architectural pattern is used in the client application. The used architectural pattern helps to separate the functional layer of the system, the model, from the user interface's aspects, which are the view and the controller. The role of each MVC's components is described as follows. The model contains the underlying classes whose instances are to be viewed and manipulated, the view contains object used to render data's appearance from the model in the user interface, and the controller has

the business logic which it controls and handles user interaction with the view and model.

## 4.2.2 Web service components

The web service components ‘Person’s Information Extraction’, ‘Camera Selection’, ‘Camera Control’, and ‘Layout Generation’ services were implemented using .NET framework. In this section we will describe and list the main technologies used by some of the web services components and discuss the Web Service interface definition of each implemented component.

### VeriLook Face Identification SDK

The ‘Face Detection and Comparison WS’ component uses VeriLook Face Identification SDK to identify and extract the features of a face from an image. The SDK enables the component to perform face feature comparison with other face feature [19].

### Web Service Interface definition

#### *Sensory Environment Metadata Repository*

**Operation:** addSensor

This service is invoked whenever a sensor is to be added to the system. Each sensor has a unique identifier that to be used in order to add/update the information, delete or disable the sensor, or to retrieve its information.

- *Input Message:* addSensorRequest

Part Name	Part Type	Description
info	SensorInfo	Information about added sensor

- *Output Message:* addSensorResponse

Part Name	Part Type	Description
result	xsd:String	Sensor identifier which to be used to retrieve sensor's information

**Operation:** removeSensor

Using the unique sensor identifier, this service removes an existing sensor form the system.

- *Input Message:* removeSensorRequest

Part Name	Part Type	Description
identifier	xsd:String	Sensor identifier

- *Output Message:* removeSensorResponse

Part Name	Part Type	Description
result	xsd:Boolean	Identifying the success or failure of the requested operation

**Operation:** getSensors

This service retrieves all the sensors stored in the database, including the disabled ones.

- *Input Message:* getSensorsRequest

Part Name	Part Type	Description
None		

- *Output Message:* getSensorsResponse

Part Name	Part Type	Description
sensorList	sensorInfo [1...unbounded]	A list containing sensors' information stored in the database

**Operation:** getSensorInformation

The 'getSensorInformation' service retrieves sensor's information. The request accepts the unique identifier of a sensor and returns its information.

- *Input Message:* getSensorInformationRequest

Part Name	Part Type	Description
identifier	xsd:String	Sensor identifier

- *Output Message:* getSensorInformationResponse

Part Name	Part Type	Description
info	sensorInfo	Sensor's information

**Operation:** updateSensorInformation

This service is invoked in order to update a sensor's information such as location, IP address, etc...

- *Input Message:* updateSensorInformationRequest

Part Name	Part Type	Description
updatedInfo	sensorInfo	Updated sensor information
identifier	xsd:String	Sensor identifier

- *Output Message:* updateSensorInformationResponse

Part Name	Part Type	Description
result	xsd:Boolean	Identifying the success or failure of the requested operation

**Non-sensory Environment Metadata Repository****Operation:** addLandmark

This service is invoked whenever a new landmark is introduced to the surveilled area. Each landmark has a unique identifier to be used in order to add/update the information, delete or disable the landmark, or to retrieve its information.

- *Input Message:* addLandmarkRequest

Part Name	Part Type	Description
info	LandmarkInfo	Information about added landmark

- *Output Message:* addLandmarkResponse

Part Name	Part Type	Description
result	xsd:String	Landmark identifier which to be used to retrieve landmark's information.

***Operation:*** removeLandmark

Using the unique landmark identifier, this service removes an existing landmark form the system.

- *Input Message:* removeLandmarkRequest

Part Name	Part Type	Description
identifier	xsd:String	Landmark identifier

- *Output Message:* removeLandmarkResponse

Part Name	Part Type	Description
result	xsd:Boolean	Identifying the success or failure of the requested operation

***Operation:*** getLandmarks

This service retrieves all the landmarks stored in the database, along with their information.

- *Input Message:* getLandmarksRequest

Part Name	Part Type	Description
None		

- *Output Message:* getLandmarksResponse

Part Name	Part Type	Description
landmarkList	LandmarkInfo [1...unbounded]	A list containing landmarks' information stored in the database

**Operation:** getLandmarkInformation

The 'getLandmarkInformation' service retrieves landmark's information. The request accepts the unique identifier of the landmark and returns its information.

- *Input Message:* getLandmarkInformationRequest

Part Name	Part Type	Description
identifier	xsd:String	Landmark identifier

- *Output Message:* getLandmarkInformationResponse

Part Name	Part Type	Description
info	LandmarkInfo	Landmark's information

**Operation:** updateLandmarkInformation

This service is invoked in order to update a landmark's information such as location, IP address, etc... The service accesses the database to update the landmark with the information provided.

- *Input Message:* updateLandmarkInformationRequest

Part Name	Part Type	Description
identifier	xsd:String	Landmark identifier
updatedInfo	LandmarkInfo	Updated landmark's information

- *Output Message:* updateLandmarkInformationResponse

Part Name	Part Type	Description
result	xsd:Boolean	Identifying the success or failure of the requested operation

### ***GPS based Location and Orientation***

#### ***Operation:*** getUserLocationOrientation

The 'getUserLocationAndOrientation' provides the requestor with the location of the user in the surveilled area, and the direction he is looking. The service accesses the GPS (through the API the GPS provides) in order to retrieve this information.

- *Input Message:* getUserLocationOrientationRequest

Part Name	Part Type	Description
identifier	xsd:String	GPS identifier

- *Output Message:* getUserLocationOrientationResponse

Part Name	Part Type	Description
xLocation	xsd:String	The X location within the region
yLocation	xsd:String	The Y location within the region
orientation	xsd:String	The current orientation of the user

### ***Media feature extraction and comparison (VeriLook [19])***

#### ***Operation:*** ExtractFeatures

The service is implemented by VeriLook, where it extracts the face feature from a specified image.

- *Input Message:* ExtractFeaturesRequest

Part Name	Part Type	Description
img	xsd:base64Binary	Image converted to base64Binary
width	xsd:Integer	The width of the image
height	xsd:Integer	The height of the image

- *Output Message:* ExtractFeaturesResponse

Part Name	Part Type	Description
ExtractFeaturesResult	xsd:base64Binary	Face feature extracted in base64Binary

**Operation:** VerifyFeatures

The ‘VerifyFeatures’ service compares two extracted face features and generates a compatibility value based on the comparison. The compatibility value is between 0 and 1. A higher value, means a higher match between two face features.

- *Input Message:* VerifyFeaturesRequest

Part Name	Part Type	Description
Feature1	xsd:base64Binary	Face feature to compare with
Feature2	xsd:base64Binary	Face feature to compare to

- *Output Message:* VerifyFeaturesResponse

Part Name	Part Type	Description
VerifyFeaturesResult	xsd:Double	The compatibility value of the face feature

**Person's Information Extraction****Operation:** getPersonInformation

This service, and with the use of the 'Media Feature Extraction and Comparison' service, compares the features extracted with the one stored in the database, and provides the requestor with the contact and personal information about the person identified within an image.

- *Input Message:* getPersonInformationRequest

Part Name	Part Type	Description
imageLocation	xsd:String	The URI location of an image

- *Output Message:* getPersonInformationResponse

Part Name	Part Type	Description
info	PersonInfo	Personal and contact information

**Camera Selection****Operation:** getDelegatedCameraNearRegion

Based on the region identified by the officer, this service selects and delegates a number of cameras to cover a subregion. The list is then used by the 'Camera Control' component to control each camera accordingly.

- *Input Message:* getDelegatedCameraNearRegionRequest

Part Name	Part Type	Description
info	regionInfo	The targeted region to cover

- *Output Message:* getDelegatedCameraNearRegionResponse

Part Name	Part Type	Description
-----------	-----------	-------------

cameraList	sensorInfo [1...unbounded]	The list of cameras delegated
------------	-------------------------------	-------------------------------

**Operation:** getDelegatedCameraNearOfficer

Based on officer's location, this service selects and delegates a number of cameras that are near to the security officer. The list is then used by the 'Camera Control' component to control each camera accordingly.

- *Input Message:* getDelegatedCameraNearOfficerRequest

Part Name	Part Type	Description
None		

- *Output Message:* getDelegatedCameraNearOfficerResponse

Part Name	Part Type	Description
cameraList	sensorInfo [1...unbounded]	The list of cameras delegated

**Camera Control****Operation:** manualCameraControl

The 'manualControlCamera' provides the feature of controlling the camera manual. The service accepts the information of the sensor that to be controlled, and the panning/tilting values.

- *Input Message:* manualCameraControlRequest

Part Name	Part Type	Description
panningDegree	xsd:Float	The panning degree angle
tiltingDegree	xsd:Float	The tilting degree angle

cameraInfo	sensorInfo	The information about the camera to control
------------	------------	---

- *Output Message:* manualCameraControlResponse

Part Name	Part Type	Description
info	streamInfo	The stream info of the camera

***Operation:*** controlCamera

Unlike the ‘manualControlCamera’, the ‘controlCamera’ service implements our proposed camera control strategy to cover a region with minimal effort.

- *Input Message:* controlCameraRequest

Part Name	Part Type	Description
info	regionInfo	The region’s information to cover

- *Output Message:* controlCameraResponse

Part Name	Part Type	Description
info	streamInfo	The stream info of the camera

***Layout Generation***

***Operation:*** getLayoutInformation

Based on the settings set by the officer using the application’s GUI, the service provides the information about the landmarks, sensors, and subregions, that to be drawn on the graphical user interface.

- *Input Message:* getLayoutInformationRequest

Part Name	Part Type	Description
settings	viewSettings	Settings set by the user

- *Output Message:* getLayoutInformationResponse

Part Name	Part Type	Description
sensorList	SensorInfo [1...unbounded]	The list of sensors to be displayed on the GUI
landmarkList	LandmarkInfo [1...unbounded]	The list of landmarks to be displayed on the GUI
regionList	RegionInfo [1...unbounded]	The list of subregions to be drawn on the graphical user interface

### *Sensory Streams*

**Operation:** controlSensor

The 'controlCamera' service controls a specific camera to pan right/left or tilt up/down.

- *Input Message:* controlCameraRequest

Part Name	Part Type	Description
direction	xsd:String	The direction to pan/tilt (right/left/up/down)
angle	xsd:Float	Panning/tilting angle

- *Output Message:* controlCameraResponse

Part Name	Part Type	Description
info	streamInfo	The stream info of the sensor

# Chapter 5

## System Performance Evaluation and Results

This chapter is divided into three sections. In the first section, the graphical user interface of the system is evaluated, where different experiments were conducted by different types of users. The 'layout view UI' component is mainly evaluated as well as the overall user interface of the system. The second section evaluates the system response time, and the last section of this chapter evaluates the proposed camera control strategy, which is also compared to our previous proposed strategy that introduces a camera control strategy using fuzzy logic approach.

### 5.1 Graphical User Interface Evaluation

The layout view of system's graphical user interface is evaluated based on experiments conducted by different users. The GUI was evaluated by 50 users, where 66 % of the users were male and 34 % were female. The age range of the participants was between 19 to 30 years old where the average age was 24 years old. Table 5.1 provides an overview

of the users that participated in the experiments.

<b>Age Range</b>	<b>Male</b>	<b>Female</b>
19 - 22	24%	16%
23 - 26	34%	18%
27 - 30	8%	0%
<b>Age 24</b>	<b>Male 66 %</b>	<b>Female 34 %</b>

Table 5.1: Overview of participants

Each user was assigned multiple tasks where these tasks were repeated for each different setup and approach proposed. A member of the evaluation team was randomly placed into five different locations in the hallway, and the user had to identify the subregion that contained the targeted person using our implemented system. The tasks were repeated for each of the two approaches proposed, for different size of subregions, and enabling/disabling the ‘landmarks view’ option in the layout view component. During the experiments, we recorded: 1) the number of times the user correctly identified a subregion, and 2) the time taken by the user to identify a subregion.

### 5.1.1 Fixed Layout display Approach (FLDA)

In this experiment, a member of the evaluation team was requested to stand at random places in the hallway, and the user had to identify the subregion in which the targeted person was located. The experiment was repeated five times for the two different settings that were available in the ‘Layout View’ component. The first setting was the subregion identification without showing the landmarks, and the second was with enabling the view of landmarks on the ‘layout view’ UI component. The time the user took to identify the sub region, and numbers of times the user correctly identified the correct region were collected.

- ***Without using landmarks***

In this case, the user were asked to identify the region in which the walking person was located without using the ‘landmarks view’ option. The view shown consisted by installed cameras, current user’s position, and the sub regions to select from.

As shown in figure 5.1, identifying a sub region with less than 1 meter was a difficult task, since only 19 % of the total number of selections made by users, identified the correct subregion. That was related to the high number of subregions that the users had to select from. Increasing the size of the subregion, decreases the number of subregions in the map view, which also increases the probability of the user selecting the correct subregion. This was also noticed for regions that are bigger than 1.50 meters where the average of correctly identified subregions was about 67 % and the time taken by the users to identify a correct subregion was about 4 seconds (see Table 5.2).

<b>Subregion dimension</b>	<b>Time taken (Using landmarks)</b>	<b>Time taken (Without using landmarks)</b>
0.50 m	5.66 sec	6.27 sec
0.75 m	5.11 sec	5.81 sec
1.00 m	4.33 sec	4.93 sec
1.25 m	3.89 sec	4.55 sec
1.50 m	3.55 sec	4.12 sec
1.75 m	3.28 sec	3.90 sec

Table 5.2: Average time taken to identify a sub region in FLDA

- ***Using landmarks***

The ‘landmarks view’ option was enabled in this experiment. The users were able to see some landmarks located in the hallway. The ‘landmarks view’ option displays some landmarks on the ‘layout view UI’ component. The users, who were involved in the previous experiment, participated in the current experiment.

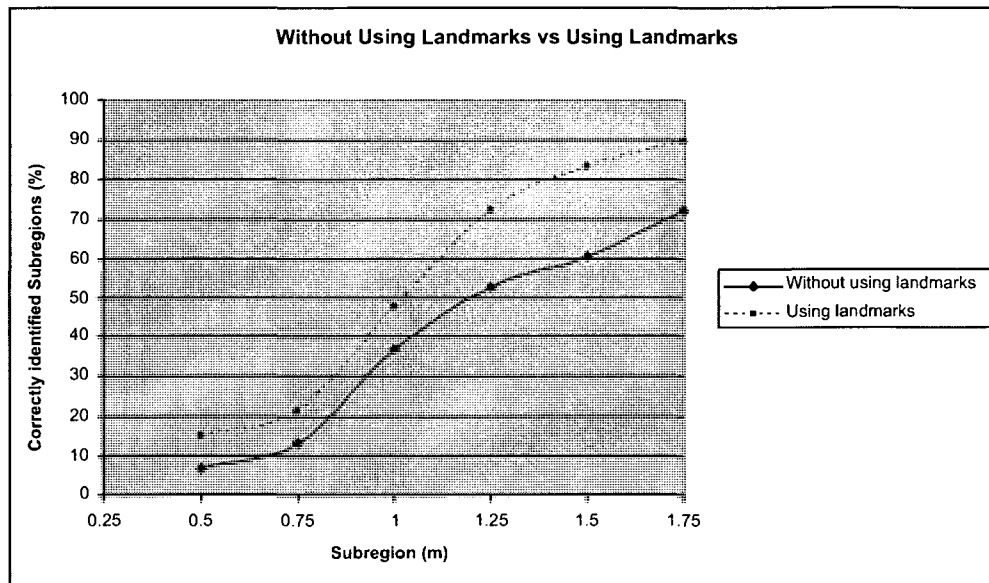


Figure 5.1: Without using landmarks vs Using landmarks - Fixed layout approach

As shown in figure 5.1, for 1.25 meter subregions, 72 % of the total selections made by users identified the correct subregions. Nevertheless, the average of correctly identifying a subregion bigger than 1.25 meter was about 81 % in 3.5 seconds average (Table 5.2). As per 1.00 meter subregion length, the percentage was not acceptable where only 49 % of selections made identified the correct subregion.

Table 5.2 and figure 5.1 clearly shows that enabling the landmarks option in the map view component, helped the users to better identify a subregion. The time the users took to identify a region using landmarks option, was lower than identifying a region without using the landmarks option. In addition to that, the percentage of correctly identifying subregions improved to 22 % for 1.5 meter subregion as shown in figure 5.1. In the following section, the same experiments conducted in fixed layout approach was conducted in the dynamic layout display approach.

### 5.1.2 Dynamic Layout display Approach (DLDA)

In dynamic layout display approach, the concept of displaying the subregions in the map view component was different than the fixed layout approach. In this approach, the subregions were drawn dynamically based on the location and direction of the security officer. Note that, the details were discussed in section 4.1, where the difference between both proposed approaches was clearly explained. As in the previous experiment, the same 50 users were asked to identify a subregion that contained a walking person using the system's GUI. Two different settings were used: 1) using landmarks, and 2) without using landmarks.

- **Without using landmarks**

As shown in figure 5.2, for 1.00 meter subregions, it is clear that 46 % of the total number of selections made by users identified the correct subregions. However, identifying subregions greater than 1.5 meter was made in less than 3 seconds (see Table 5.3), where the percentage of correctly identifying the correct subregion was about 75 %. Table 5.3 shows clearly the time taken by users to identify a subregion using both options: using landmarks, and without using landmarks.

<b>Subregion dimension</b>	<b>Time taken (Using landmarks)</b>	<b>Time taken (Without using landmarks)</b>
0.50 m	4.23 sec	4.76 sec
0.75 m	3.41 sec	4.11 sec
1.00 m	3.11 sec	3.83 sec
1.25 m	2.72 sec	3.29 sec
1.50 m	2.75 sec	2.96 sec
1.75 m	2.69 sec	2.89 sec

Table 5.3: Average time taken to identify a sub region in DLDA

- **Using landmarks** As shown in figure 5.2, users were able to identify 90 % of 1.5 and 1.75 meter subregions by 2.72 seconds in average. However, identifying 1 meter

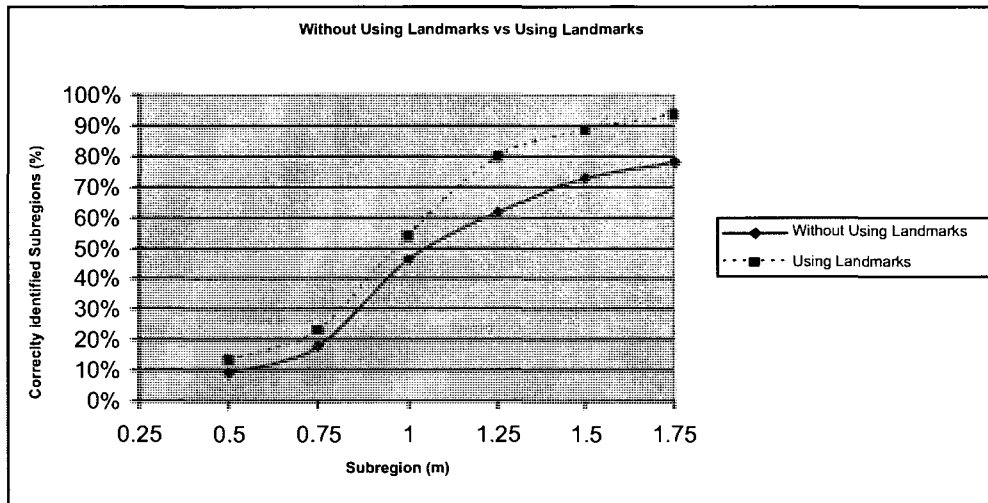


Figure 5.2: Without using landmarks vs Using landmarks - Dynamic layout approach

and smaller subregions is still not acceptable, since users were only able to identify 54 % and 22.40 % of 1 and 0.75 meter subregions.

In conclusion, same observation observed in Fixed layout display approach was observed here. Introducing the landmarks view option in the layout view component helped the user to easily identify the regions. However in dynamic layout approach, it was noticeable that the users took less time to identify the correct subregions than fixed layout approach.

### 5.1.3 Comparison between FLDA and DLDA approaches

From the results obtained from the two different approaches, enabling the landmarks view has a great impact on users' decision in selecting a subregion. However, in both approaches, users had difficulty in identifying subregions less than 1 meter. In other hand, the percentage had increased from 60 to 74 % for subregions greater than 1 meter, when using the landmarks view option (see figures 5.1 and 5.2). When introducing the landmarks view option, most of the users could relate the location of a nearest landmark

to the targeted person's position, which helped them in identifying the subregion of the target.

As pointed in Tables 5.2 and 5.3, the time the users took to identify the region was slightly better in the dynamic layout display approach. However, as we were running the experiments, we noticed that 22 % of the users were more comfortable in identifying subregions using the fixed layout view, where the majority of the users preferred the use of dynamic layout view, and that reflected the results shown in the above tables about the time they took for both approaches. We expected that the users could easily identify bigger/wider subregions such as 1.5 meter or 1.75 meters in both approaches, but we observed that they had some difficulty or in doubt in identifying a region when a person in the hallway is standing at the border of a subregion using the fixed layout approach.

In conclusion, the dynamic layout display has slightly proven that is easier to deal with in identifying a subregion, and that could be shown in Table 5.3 and figure 5.2 where for sub regions wider than 1.25 meters, the percentage of correctly identifying the subregion was about 84% in 2.75 seconds, where it was 74% in 3.7 seconds for the fixed layout display approach, as shown in Table 5.2 and figure 5.1.

## **5.2 System Response Time**

In this experiment, we evaluate the system performance and response time. During the experiments conducted, we have captured in Table 5.4 the time the system took to display the pictures of the scene after a subregion selection, to retrieve the information of a captured user, and to draw the layout on the map view component. Note that, displaying the pictures of the scene require the cameras to be controlled, however in Table 5.4, we have excluded the time the camera took to cover a region, since each camera has different settings, and might have different panning/tilting speed.

Category	Description of test category	Response Time
1	<i>Display the pictures of the scene after a subregion selection</i>	2.03 sec
2	<i>Retrieve the information of a captured user</i>	1.29 sec
3	<i>Draw the layout view</i>	
	4 subregions	0.95 sec
	9 subregions	1.33 sec
	16 subregions	1.74 sec
	32 subregions	2.35 sec

Table 5.4: System Response Time

All test categories listed in Table 5.4 requires multiple SOAP requests to be invoked to fulfill the task required. Because the officer will be moving from place to place, the layout view must be updated every 2-5 seconds since the layout view depends on the current location and orientation of the officer. Drawing 32 subregions, landmarks, and cameras installed, the system requires 2.35 seconds in average to get the current location of the officer, get nearest installed cameras and landmarks, and form the SOAP response that includes the information for all subregions and landmarks to be drawn. The time also includes the processing time the application needs to process SOAP response in order to draw the sub regions. As shown in Table 5.4, the time is increasing whenever the number of subregions is increasing. SOAP response includes information about all subregions to be drawn, and that has an effect on the response time of the system since increasing the number of subregions requires much effort by the system to process the request and the response. Updating the subregions of map view component is time consuming, however, this could be enhanced by updating only the subregions that are newly introduced to the map instead of updating all subregions of a layout view. On the other hand, we can neglect the response time the system took to display the pictures of the scene and the time took to retrieve the information which is about 2.03 and 1.29 seconds respectively,

since these tasks are required and invoked once for the whole process of retrieving the information of a captured person.

Finally, most of the web services were running on a single core processor machine (P4 2.4 GHz, 2GB Ram). However, the response time of all tasks in Table 5.4 could be enhanced by running the web services on a more powerful machine like dual core or quad core processor machine.

### **5.2.1 Overall User Satisfaction**

In this experiment, the overall user interface was evaluated, as well as the two proposed layout approaches. Each of the 50 users who were involved in previous experiments were given some tasks in order to test all the functionalities of the system.

These tasks were as follows:

1. Manually control a camera in the Camera View Component
2. Switch between cameras' view
3. Control a camera to cover an object
4. Select another camera for a view display
5. Identify a sub region that contains a person
6. Select and identify the image and person within the image
7. Scan through personal information
8. Add activity comment

A questionnaire survey in Appendix A was provided to users upon completion their tasks. Through the surveys we analyzed the usability, understandability, and the improvement that could be done. Table 5.5 shows the analysis based on the surveys and feedbacks provided.

<b>Actions performed</b>	<b>% Users consider it simple</b>	<b>% Users consider it complex</b>
Activate a camera	84 %	16%
Control a camera	56 %	44 %
Select a camera from the map view	72 %	28 %
Change map view settings	100%	-
Identify a subregion	78 %	22 %
Scan through captured persons info	90 %	10 %
Add activity comment	86 %	14 %
Select a picture of a scene	96 %	4%
<b><i>Average</i></b>	<b>82.75 %</b>	<b>17.25 %</b>

Table 5.5: Overall System usability

From Table 5.5, it is evident that 83 % of the users find the system easy to use, and that the tasks and operations performed are within the capability of any user who has previous experience in using a PC, and does not require expert to complete each task. We can also conclude that even when enabling landmarks view, identifying a subregion is the biggest and toughest challenge. However, identifying a region is one of the major and important decisions that should be properly taken by the user, so that the output of the system would be effective. Later in this section, we will provide and discuss list of suggestions that could improve the output of the system.

For the tasks that are related in identifying a subregion, table 5.6 compares the fixed and dynamic layout approach in terms of understandability, and the amount of time needed to identify a subregion.

	Dynamic Layout Approach	Fixed Layout Approach
Layout Display Choice	78 %	22%
Time needed to identify a subregion	3.39 sec	4.61 sec
Correctly identified region (Subregion 1-1.5 m. Using landmarks)	74.00 %	67.73 %

Table 5.6: Dynamic vs Fixed Layout Approach

Dynamic layout approach dominates the fixed layout approach in terms of the percentage of correctly identified subregions, and time taken for identification, where 74 % of the 1 to 1.5 meters subregions could be correctly identified in 3.39 seconds. The dynamic also dominates the fixed approach in terms of users preferences where 78 % of the users preferred using the dynamic approach in identifying a subregion.

The following are couple of issues and suggestions that were provided by users who thought that they could improve the effectiveness of our overall.

- **Camera activation:** In the camera view component, the camera that is displayed on the main view could be manually controlled by the user. In order to control the other cameras, users had to activate one of them. This additional task had delayed some of the users from performing the action.

#### Suggestion

- Give the user the ability to control each of the three cameras shown on the camera control view, in other words, include a control panel for each camera without having the option to activate one of the cameras and control it.
- **Manual camera control:** To control a camera in order to cover a specific region, the user has to activate a camera and control it using the arrow buttons. These buttons pan/tilt the camera by a specified degree. To cover a region, the users had

to repetitively press the button until the targeted region was covered.

#### **Suggestion**

- The camera should continually pan/tilt until the user removes the press on a button
- **Shape and size of the landmarks:** Using landmarks has good effect on correctly identifying a region. However, the landmarks shown in the map view, does not reflect the actual size and shape of the landmark that it represents, which also lead to confusion to some of the users.

#### **Suggestions**

- Have the landmarks reflect the actual size, shape, and color they represent.
- Providing a 3-Dimension view, which could improve the % of identifying a region

### **5.3 Camera Control Evaluation**

In this section, the camera control strategy proposed is evaluated. Different settings and setup were used, where the purpose of the experiments was to study the efforts made by cameras, the time the cameras took to cover a region, and the quality of the pictures taken by the cameras. The same installation setup was used for all three experiments. Three dynamic (PTZ) cameras were installed in a 5.40 x 14.40 meters wide rectangular area. The cameras were located at the height of 2.70 meters. Two cameras were installed on one side of a wall and the third one on the opposite side of the wall as shown in figure 5.3. Initially, each camera was set to its idle position where the panning angle of each camera was zero.

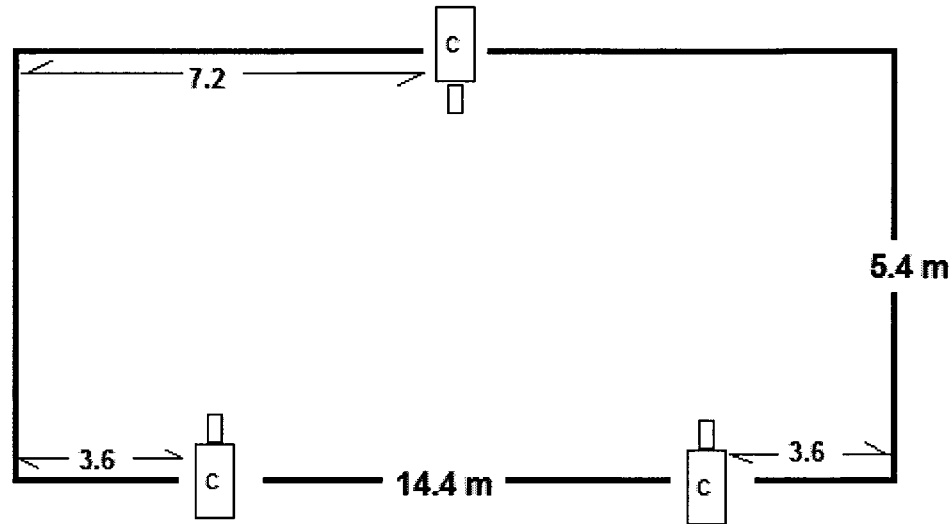


Figure 5.3: Laboratory setup

In the first experiment, the efforts of the three cameras to cover a subregion are evaluated using different subregion dimensions. However, in the second experiment, for each subregion dimension, the percentage of the pictures identified is determined which represents the quality of the pictures taken by the cameras. We mean by an identified picture, is a picture that contains the targeted subject.

### 5.3.1 Experiment 1: Evaluating Cameras' Effort

All installed PTZ cameras were controlled to cover the regions that to be selected by users. In this experiment, 50 random subregions were defined by the evaluation team and used to determine the efforts of the three cameras installed. The six different subregion dimensions used to determine the efforts were: 0.75 m, 1.00 m, 1.25 m, 1.50 m, and 1.75 m. Figure 5.4 illustrates the average efforts in degrees for the three cameras for each subregion dimension.

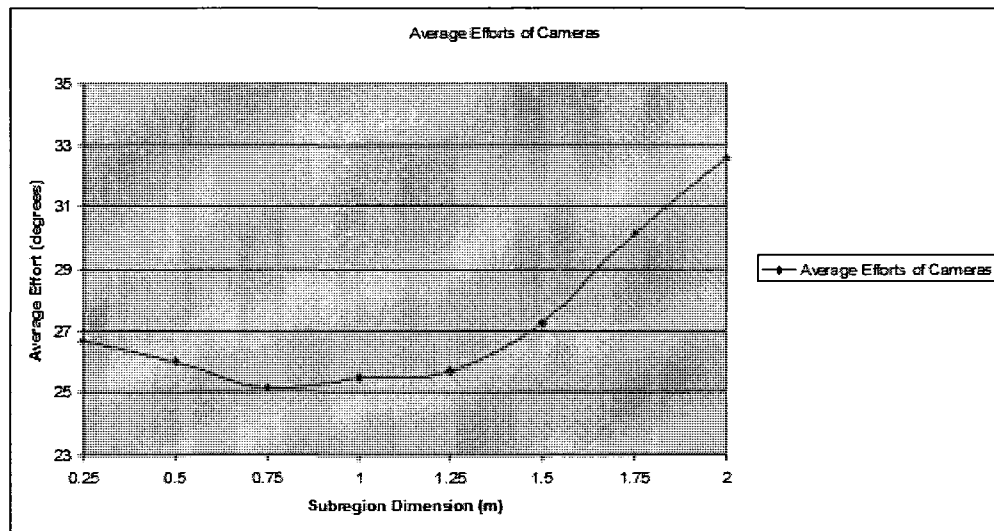


Figure 5.4: Average efforts of the three installed cameras

Cameras make more efforts when covering small subregion as well as covering bigger subregion. As shown in figure 5.4, the effort decreases when the dimension increases for subregions small than 0.75 meter, however the effort significantly increases for subregion dimension greater than 1.25 meters. Based on the setup in the lab, and the results obtained, the ideal effort for the cameras is 26 degrees for a subregion dimension between 0.6 to 1.3 meters. However, figure 5.5 shows the subregion coverage percentage by the cameras, where each of the four lines represents the percentage coverage of a subregion of different dimension.

The initial coverage percentage for subregions 1.00 to 1.50 meters was about 34 % of the area of a subregion. On the other hand, the initial percentage of smaller regions is about 50 % where the initial coverage of 0.25 m subregion is 53 % as illustrated in figure 5.5. From figure 5.5, we conclude that the coverage percentage slightly increases when the effort of the cameras reaches up 15 degrees, however it significantly increases for 1 to 1.5 meters subregions comparing to the smaller subregions. Note that, the above

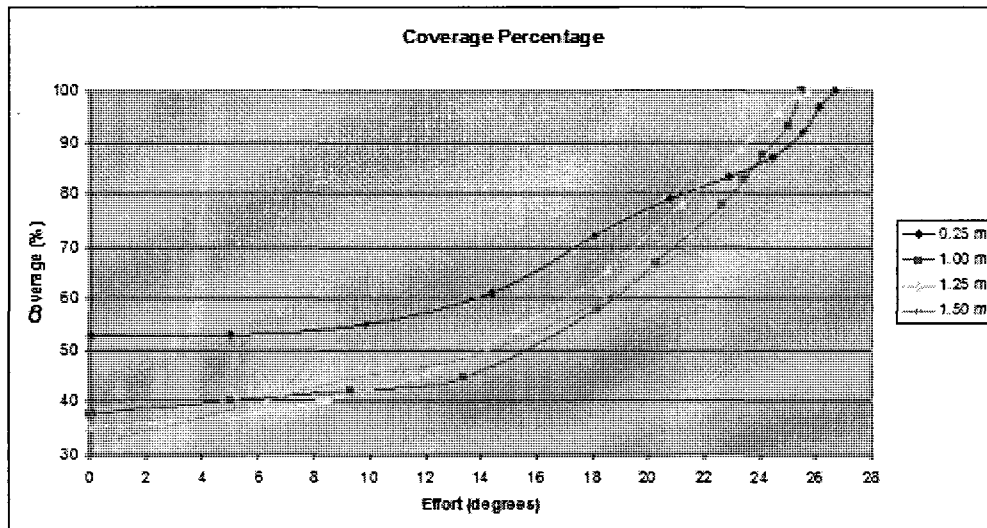


Figure 5.5: Subregion Coverage Percentage

observation is deduced in our specified setup and can not be generalized. However, for other setups, the proposed setup can be used to derive such observation.

### 5.3.2 Experiment 2: Quality of Scene's Pictures

In the previous experiment, the efforts of the cameras were evaluated. However, in this experiment we analyzed the quality of the picture taken by the cameras to cover the scene. A member of the evaluation team was asked to move in hallway. By identifying the subregion that contains the walking person and the direction of that person, the security system was enabled to send commands to the cameras to make three attempts to take pictures of scene. In the first attempt, the cameras take picture of the scene based on the subregion identified by the user, and for the second and third attempt, the system predicts the targeted person's location in the hallway (based on the direction identified by the user) to send command to the camera to cover the predicted subregion. After each attempt, three pictures were sent back to the user for evaluation. These

pictures were classified into one of the two categories: 1) identified target, 2) unidentified target. In other words ‘identified target’ category of a picture means that the target is identified within the picture by the security personnel. The experiment was repeated for each subregion dimension. The walking person was asked to be placed in 30 different subregions in each setup, where thirty pictures were taken by each camera per each attempt, 90 pictures were taken in total per camera in each attempt. Table 5.7, shows the percentage of identified target in each attempt and in each setup.

<b>Setup</b>	<b>Attempt 1 % pic. identified</b>	<b>Attempt 2 % pic. identified</b>	<b>Attempt 3 % pic. identified</b>	<b>% pic. un- identified</b>
0.50 m subregion	12.22 % (11)	17.77 % (16)	18.88 % (17)	51.11 %
1.00 m subregion	21.11% (19)	27.77 % (26)	32.22 % (29)	17.77 %
1.25 m subregion	23.33 % (21)	33.33 % (30)	27.77 % (25)	15.55 %
1.50 m subregion	23.33 % (21)	35.55 % (32)	31.11% (28)	10.00 %
1.75 m subregion	26.66 % (24)	40.00 % (36)	23.33 % (21)	9.00 %
2.00 m subregion	27.77 % (25)	36.66 % (33)	27.77 % (25)	7.00 %

Table 5.7: Percentage of images that identified the target taken within three attempts

For 0.5 meter subregion setup, only 12 % of the pictures in which security officer could identify the target and 51 % of the pictures were classified as unidentified. That was related to the factor to the dimension of the subregion which lead the person to be moved to another region and the cameras missed the target. On the other hand, for subregions 1.25 and 1.5 meters, 87 % of the pictures could be identified where about 60 % of the pictures were identified in the first two attempts. However, for the last two setups, the results were unexpected since we didn’t notice any difference between the percentage in setup 3 and 4 and the percentage in setup 5 and 6. Our assumption was wrong when

we thought that when having a wider region, the camera was more capable to capture the target, but we noticed that the cameras required more efforts to cover wider regions which resulted in missing the target, and another observation was that when the person was moving to the opposite panning direction of the camera also resulted in missing the target.

Figure 5.4 shows that subregion of 0.75 meter requires minimal efforts to cover a subregion. However, for a 0.75 meter subregion, only 22.40 % of the subregions could be correctly identified by the user as shown in figure 5.2. On the other hand, considering 1.25 meter subregions, up to 80 % of the subregions could be correctly identified with slightly more effort required by camera comparing to 0.75 meter subregion. In addition to that, results showed that up to 85 % of the pictures taken for 1.25 meter subregion could clearly identify the target after the third attempt. In conclusion, users were more comfortable when selecting 1.25 meter subregions and it was efficient for the installed cameras since it required less effort than subregions greater than 1.25 meter and less than 0.75 meter.

### **5.3.3 Comparison to fuzzy logic approach camera control**

In our previous work [4], a 'Camera Control' approach was proposed using fuzzy logic. An experiment was conducted in a two meter wide hallway and three cameras were installed at a height of 2.5 meters. The purpose of the experiment is to test the accuracy of the decision taken by the fuzzy system. For each test, a volunteer was asked to be placed at a random place in the hallway where the fuzzy inference system activates one of the cameras and controls it to take a face image of the volunteer. The image is then classified good or bad image based on the face extraction and comparison algorithm used

[20]. For the purpose of this experiment, only three fuzzy input variables were used for controlling the camera. Since it is in the first stage of the application, we assumed that the person is standing in the hallway and not moving.

The fuzzy input variables used are:

1. 'Location' variable: The position of the passenger in the hallway (e.g. on the left side of the hallway)
2. 'DistanceFromCam' variable: The distance between the passenger and one of the installed cameras
3. 'Height' variable: The height of the passenger

The 'Location' and 'DistanceFromCam' fuzzy variables have three fuzzy values each. Left, Middle, and Right values for 'Location' variable, and Near, Average and Far values for 'DistanceFromCam' input variable. The 'height' input has five fuzzy values which are Very Short, Short, Average, Tall, and Very Tall. Using these three inputs, the fuzzy system is able to determine three fuzzy outputs needed for the camera so that it can move to get a good face image for the passenger.

The three output variables are:

1. 'Camera' variable: The camera that is most likely to take a good face image
2. 'LRangle': The angle needed for the selected camera to rotate to the right or to the left
3. 'TDangle' variable: The angle needed for the selected camera to move up or down

The input data were classified into six categories as shown in Table 5.8.

Category Type	Name	'Location' variable
A	Very Far Left	Near the left hall of the hallway
B	Far Left	On the left side of the hallway
C	Middle	In the middle of the hallway
D	Far Right	On the right side of the hallway
E	Very Far Right	Near the right hall of the hallway

Table 5.8: Input test data categories

Based on the results obtained, in average 79.5 % of picture could be identified as good images, where in our camera control strategy proposed in the thesis, 87 % of the picture could be classified as good image of 1.25 - 1.5 meters subregions.

## Chapter 6

# Conclusion and Future Work

A proactive visual surveillance system for public security was proposed where three main contributions were addressed. The system's architecture, system's desktop application (graphic user interface), and the camera control strategy.

The proposed architecture introduces the concept of web services technologies which share data, process, and business logic across the network. Previous contributions have been made in measuring the performance of web services technologies, and these contributions show that web services-based application scales and performs well. In addition to that, re-usability of existing components is another advantage of using web services which allows developers for easily integration during development phase.

Human-cum centric approach was proposed to allow users in assisting the system in identifying a suspected person. Designing a usable user interface helps the user to easily interact with the system which in general would affect the effectiveness of the overall system's output. The graphical user interface was divided into four main components and each of the components was evaluated. The results show that the user interface was classified as easy to use by 83 % of the users who participated in the experiments. However, decreasing the amount of time required by the user to select a subregion, will

also affect the system's effectiveness output. To address this issue, different map display approaches were proposed. The two display approaches proposed are the dynamic layout approach and the fixed layout approach. 78 % of the users that participated in the experiments preferred using the dynamic layout approach rather than using the fixed layout approach, where up to 88 % of the subregions selection could be correctly identified by users and the average amount of time took in identifying a 1.25 - 1.5 meter subregion is 3.15 seconds.

The camera control strategy proposed is a simple strategy which computes the minimal effort required by each camera to cover a subregion. The results show that 0.75 to 1.25 meter subregions require the minimum efforts (25 - 26 degrees) by the cameras to cover the subregion, and up to 85 % of the picture captured could be identified for 1.25 meter subregion where the targeted person has been captured within the image.

We have noticed during the experiments that updating the layout view on the 'Map view' component was time consuming. This issue should be taken into consideration in the future for a better system performance. In addition to that, capturing a face image in a crowded environment is a challenging task, where some possible researches could be on tracking an identified person in a crowded place for a better face image capturing. In our system, VeriLook library was used to extract face feature of an image and compare it with another. However, the extraction and comparison does not work effectively where the user is too far from the camera, where some work could be applied to provide more powerful algorithms to provide a better and more reliable library that would work in crowded places. Some work could be also applied to the application's GUI. In our system, selecting a sub region is a difficult task for the user. This could be enhanced by displaying the map in a 3D, and displaying installed landmarks as they appear.

## Appendix A

### Questionnaire for User Interface Satisfaction

1- How do you rate the system in terms of usability	hard	1	2	3	4	easy
a. Activate a camera from the "Camera View" component		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
b. Select a camera from the "Map View" component		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
c. Manually control a camera		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
d. Identify a subregion in the "Map View" component		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
e. Select a picture of a scene		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
f. Scan through captured person's info		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
g. Add activity comment		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

2- How do you rate the screen view		1	2	3	4	
a. Reading characters on the screen	hard	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy
b. Organization of information	confusing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	very clear
c. Sequence of screens	confusing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	very clear

3- Layout display view		Yes	No	Somewhat
a. Does enabling landmarks view feature view has impact on selecting a subregion?		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b. Which layout display approach do you prefer for subregions identification?		Dynamic layout <input type="radio"/>	Fixed layout <input type="radio"/>	No preferences <input type="radio"/>

4- List the most negative aspect(s)
a. _____
b. _____
c. _____

5- Suggestion(s)
a. _____
b. _____
c. _____

# Bibliography

- [1] ADVISOR. Advisor specification documents. (internal classification 2001).
- [2] P.K. Atrey, M.A. Hossain, and Abdulmotaleb El Saddik. Automatic scheduling of cctv camera views using a human-centric approach. *IEEE International Conference on Multimedia and Expo*, pages 325–328, 2008.
- [3] Bassem Barake, Pradeep K. Atrey, Jiyong Zhao, and Abdulmotaleb El Saddik. A user friendly mobile device interface to assist security personnel in physical monitoring at public places. In *IEEE International Conference on Multimedia and Expo, ICME 2009*, July 2009. Under review.
- [4] Bassem Barake, Jiyong Zhao, and Abdulmotaleb El Saddik. Towards an intelligent tele-surveillance system for public transport areas: Fuzzy logic based camera control. In *Canadian Conference on Electrical and Computer Engineering, 2008*, pages 001987–001990, May 2008.
- [5] R. Collins, A. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):745–746, 2000.
- [6] Microsoft Corporation. Microsoft speech technologies. World Wide Web (WWW), <http://www.microsoft.com/speech/speech2007/default.msp>.

- [7] DETEC. Leading digital detection and surveillance systems. World Wide Web (WWW), [http://www.detec.no/sider\\_eng/start.htm](http://www.detec.no/sider_eng/start.htm).
- [8] Abdulmotaleb El Saddik. Performance measurements of web services-based applications. *IEEE Transaction on Instrumentation and Measurement*, 55(5):1599–1605, 2006.
- [9] Ahmed Elgammal. Human-centered multimedia: Representations and challenges. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 11–18, Santa Barbara, California, USA, 2006.
- [10] Christopher Ferris and Joel Farrell. What are web services? *Communications of the ACM*, 46(6), Jun 2003.
- [11] Paul Fremantle, Sanjiva Weerawarana, and Rania Khalaf. Enterprise services. *Communications of the ACM*, 45(10), Oct 2002.
- [12] Yueyu Fu and Javed Mostafa. Toward information retrieval web services for digital libraries. In *Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries, 2004*, pages 370–371, Tucson, Arizona, USA, 2004.
- [13] Jeffrey Gortmaker, Marijn Janssen, and Ren W. Wagenaar. The advantages of web service orchestration in perspective. In *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, March 2004.
- [14] Gotcha. Video surveillance motion detection software for the personal computer. World Wide Web (WWW), <http://www.gotchanow.com>.
- [15] Weiming Hu, Tieniu Tan, Liang Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004.

- [16] N. Jinaporn, S. Wisadsud, P. Nakonrat, and A. Suriya. Security system against asset theft by using radio frequency identification technology. In *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008*, pages 761–764, 14-17 May 2008.
- [17] I. Junejo, X. Cao, and H. Foroosh. Geometry of a non-overlapping multi-camera network. In *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, pages 43–48, Nov 2006.
- [18] A. Lipton, H. Fujiyoshi, and R. Patel. Moving target classification and tracking from real-time video. In *Fourth IEEE Workshop on Applications of Computer Vision, 1998*, pages 8–14, Oct 1998.
- [19] C. Micheloni, G.L. Foresti, and L. Snidaro. A network of co-operative cameras for visual surveillance : Intelligent distributed surveillance systems. In *IEE proceedings. Vision, image and signal processing*, volume 152, pages 205–212. Institution of Electrical Engineers, Stevenage, ROYAUME-UNI (1994-2006) (Revue), 2005.
- [20] Neurotechnologija. Webcam capable multiplatform face identification sdk: Verilook 3.1 standard sdk. World Wide Web (WWW), [http://www.neurotechnologija.com/vl\\_sdk.html](http://www.neurotechnologija.com/vl_sdk.html).
- [21] I. Pavlidis, V. Morellas, P. Tsiamyrtzis, and S. Harp. Urban surveillance systems: From the laboratory to the commercial world. In *Proceedings of the IEEE*, pages 1478 – 1497, Oct 2001.
- [22] Physorg. Bat inspires space tech for airport security. World Wide Web (WWW), <http://www.physorg.com/news7210.htm>, 2005.

- [23] A. Pozzobon, G. Sciutto, and V. Recagno. *Security in ports: the user requirements for surveillance system*. Kluwer Academic Publishers, 1998.
- [24] R. Rangaswami, Z. Dimitrijevic, K. Kakligian, E. Chang, and Y. Wang. The sfinx video surveillance system. In *IEEE Conference on Multimedia and Expo*, pages –, Jun 2004.
- [25] Supriya Rathod. Will computers replace human intelligence? World Wide Web (WWW).
- [26] Hossein Saiedian and Shawn Mulkey. Performance evaluation of eventing web services in real-time applications. *IEEE Communications Magazine*, 46(3):106–111, 2008.
- [27] M. Shah, O. Javed, and K. Shafique. Automated visual surveillance in realistic scenarios. *IEEE Multimedia*, 14(1):334–352, Jan-Mar 2007.
- [28] Changsong Shen, Chris Zhang, and Sidney Fels. A multi-camera surveillance system that estimates quality-of-view measurement. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, pages 193–196, Sep-Oct 2007.
- [29] C. Shu, A. Hampapur, M. Lu, L. Brown, J. Connell, A. Senior, and T. Yingli. Ibm smart surveillance system (s3): a open and extensible framework for event based surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005. AVSS 2005*, pages 318–323, Sep 2005.
- [30] Vivek K. Singh and Pradeep K. Atrey. Coopetitive visual surveillance using model predictive control. In *The 3rd ACM International Workshop on Video Surveillance and Sensor Networks in conjunction with ACM Multimedia 2005 (ACM VSSN'05)*, pages 149–158, Singapore, Nov 2005.

- [31] Vivek K. Singh, Pradeep K. Atrey, and Mohan S. Kankanhalli. Cooperative multi-camera surveillance using model predictive control. *Machine Vision and Applications, Special Issue Paper*, 2007.
- [32] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999*, pages 247–252, Fort Collins, CO, USA, 1999.
- [33] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. In *IEE Proceedings - Vision, Image and Signal Processing*, volume 152, pages 192–204, Apr 2005.
- [34] S.A Velastin, L. Khoudour, B.P.L. Lo, J. Sun, and M.A. Vicencio-Silva. Prismatic: a multi-sensor surveillance system for public transport networks. In *IEE International Conference on Road Transport Information and Control, 2004. RTIC 2004. 12th*, pages 19–25, Apr 2004.
- [35] W3C. Soap specifications. World Wide Web (WWW), <http://www.w3.org/TR/soap/>.
- [36] W3C. Web services description language (wsdl). World Wide Web (WWW), <http://www.w3.org/TR/wsdl>.
- [37] Wikipedia. Web service. World Wide Web (WWW), [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service).
- [38] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland. Pfindex: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

- [39] J. Xinding Sun Foote, D. Kimber, and B.S. Manjunath. Region of interest extraction and virtual camera control based on panoramic video capturing. *IEEE Transactions on Multimedia*, 7(5):981– 990, 2005.