TAILGATING/PIGGYBACKING DETECTION SECURITY SYSTEM

CHAN TJUN WERN

MASTER OF ENGINEERING SCIENCE

FACULTY OF ENGINEERING AND
GREEN TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN
JULY 2013

TAILGATING/PIGGYBACKING DETECTION SECURITY SYSTEM

By

CHAN TJUN WERN

A dissertation submitted to the Department of Electronic Engineering,
Faculty of Engineering and Green Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Engineering Science
July 2013

ABSTRACT

TAILGATING/PIGGYBACKING DETECTION SECURITY SYSTEM

Chan Tjun Wern

Electronic access control is a system which enables the authority to control and restrict access to a target sensitive area. However, its effectiveness depends on the proper usage of the system by those who are granted access. One of the biggest weaknesses of electronic access control is the lack of a system to prevent a practice known as “tailgating” or “piggybacking”. This research plans to tackle this security issue by using video analytics technology. Traditionally, video analytics is implemented on desktop computers which have large amount of memory resources. However, this research aims to implement the tailgating/piggybacking detection security system on an embedded system with limited memory resources. The detection system developed for this research consists of two main components, a single inexpensive internet protocol camera and an embedded based control unit. To extract moving object, background subtraction with real time background update is used in the developed system. The extracted image will then undergo connected component analysis to improve its image quality. To detect tailgating and piggybacking event, a three stage violation checking algorithm is implemented in the system. The results showed that the developed system is able to detect tailgater or piggybacker successfully in various situations and can be implemented on embedded platform for smooth real time analysis.

ACKNOWLEDGEMENT

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I offer my sincerest gratitude to my supervisor, Dr. Yap Vooi Voon and co-supervisor, Dr. Soh Chit Siang who has supported me throughout my research with their patience and knowledge whilst allowing me the room to work in my own way.

I would also like to express my deepest appreciation to ELID Sdn. Bhd. for their support in this project especially Mr. H.T. Tan from R&D Department for providing technical consultation.

I am also indebted to my friends and colleagues who have assisted me from the building of the detection system until the testing and verification of the system performance. I would also like to thank them for helping me getting through the difficult times, and for all the emotional support, entertainment, and caring they provided.

Finally, I am grateful to all parties who have directly or indirectly gave their fullest co-operation to ensure a successful completion of my research.

FACULTY OF ENGINEERING AND GREEN TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _____________

SUBMISSION OF DISSERTATION

It is hereby certified that ***Chan Tjun Wern*** (ID No: ***10AGM07483)*** has completed this dissertation entitled "Tailgating/Piggybacking Detection Security System" under the supervision of Dr. Yap Vooi Voon (Supervisor) from the Department of Electronic Engineering, Faculty of Engineering and Green Technology, and Dr. Soh Chit Siang (Co-Supervisor) from the Department of Electronic Engineering, Faculty of Engineering and Green Technology.

I understand that University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Chan Tjun Wern*)

APPROVAL SHEET

This dissertation entitled "**TAILGATING/PIGGYBACKING DETECTION SECURITY SYSTEM"** was prepared by CHAN TJUN WERN and submitted as partial fulfillment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman.

Approved by:

____

(Asst. Prof. Dr. YAP VOOI VOON)
Date:
Assistant Professor/Supervisor
Department of Electronic Engineering
Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

____

(Asst. Prof. Dr. SOH CHIT SIANG)
Date:
Assistant Professor/Co-supervisor
Department of Electronic Engineering
Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name:______________________

Date:_______________________

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BSD	Berkeley Software Distribution
CCTV	Close circuit television
DSP	Digital Signal Processor
FPS	Frames per second
IP	Internet protocol
LAN	Local area network
MATLAB	Matrix Laboratory
MIPS	Million instructions per second
MJPEG	Motion JPEG
OpenCV	Open Source Computer Vision Library
PETS	Performance Evaluation of Tracking and Surveillance
RAM	Random-access memory
ROI	Region of Interest
RTSP	Real Time Streaming Protocol
US	United States
WEP	Wired Equivalent Privacy
WPA2	Wi-Fi Protected Access II

CHAPTER 1

INTRODUCTION

1.1 Access Control and Problems

Access control is a system which enables the authority to control and restrict access to a target sensitive or secured area. Access control can be found commonly at private places such as residential area or office. By denying access to unauthorized personnel, properties inside the secured area can be safeguarded.

The popularity and affordability of computer has led to the rise of electronics access control. This system grants access automatically based on the credential presented. Traditionally, access credential is a physical key used to unlock a door. However, for electronic access control, credential can be many things ranging from pin code to fingerprint as long as it is something that the user know or possess. When access is granted, the door is unlocked for a predetermined time and when access is refused, the door remains locked. The entire successful and denied entry and exit log can be recorded and stored in a database if needed; such is the advantage of having an electronic access control.

The effectiveness of electronics access control however, depends on the proper usage of the system by those who are granted access. These individuals are in control of the door from the time it unlocks until it relocks. Most of the available systems do not have control on the amount of people entering a secured area when a valid credential is presented. Once a door is opened by an authorized person, anyone can follow behind to access the restricted area; similarly, it is also very easy for intruder to exit the building with the same method. One of the biggest weaknesses of automated access control systems is the absence of a system to prevent this practice better known as “tailgating” or “piggybacking”. Tailgating is a situation where an individual follows an authorized person into the secured area without the knowledge of that authorized person. Piggybacking on the other hand occurs when a person access the restricted area with the permission of an authorized person. Tailgating and piggybacking are two serious and well-recognized security risks. A study by United States (US) government investigators (Kettle, 1999) shows that undercover agents from the US federal aviation administration repeatedly breached security measures at major airports with a 68% success rate and one of the methods used was by following airport and airline staff through the door into controlled area. The addition of tailgating/piggybacking detection system is crucial to ensure access is only granted to people with authorization.

1.2 Anti Tailgating/Piggybacking System and Weaknesses

One of the common solutions to tailgating/piggybacking problem is by installing physical barrier at the entrance such as mechanical turnstile or security revolving door. Physical barrier is well proven, effective and it is readily accept by most of the users. Typically, the barrier will be constantly attended by a security officer. The downside in using physical barrier is that it is obtrusive in appearance. The premise will also need to have a separate door or gate for emergency exit because the barrier will slow down crowd clearance during any event of emergency. In addition, it is also not handicapped user friendly. For example, disable person sitting on a wheel chair will have problem passing through a normal size physical barrier; a special wider physical barrier will be needed. With physical barrier, loading and unloading of large object will also be a problem.

Due to the weakness of the traditional solution, several new tailgating/piggybacking detection systems were developed. One of them is by using infrared break-beam system. This system works by counting the amount of people passing through the infrared beam. When a person passes through, the infrared beam will be interrupted and the system will identify this. However, this system can be easily defeated and has many shortcomings. For example, if multiple people pass through the break-beam pair at the same time, the system will fail to identify this. Another easy way to bypass this detection system is by crawling under or jumping over the break-beams. Since the break-beam requires a light source directly opposite the detector, the break-beam can be affected by the swing of a door and will cause the system to wrongly detect the door swing as a person passing through; modifications to the existing setting may thus be required for installation. Furthermore, the optical break-beams may not work well in environment with high ambient lighting conditions (Bramblet et al., 2008).

There is also an advance tailgating/piggybacking detection system that is based on 3-dimensional machine vision. This system can detect human and differentiates them from carts or other objects accurately by using 3D images generated by the stereo camera which provide a clear and detailed view of the surveillance area. Due to the sophisticated system used, the cost of installing this system is also significantly higher than other tailgating/piggybacking detection methods. A complete mantrap system with stereo vision technology will cost approximately 50,000USD (McCormick, 2007).

1.3 Objectives of Research

In view of the various shortcomings of existing solutions, a better way to prevent tailgating/piggybacking problem is by developing a detection system using video analytics technology. Video analytics is an emerging technology where computer vision is used to perform different tasks by analysing the video feed. It is widely used in applications such as traffic monitoring, human action recognition and object tracking. This technology can reduce the work load of a human operator and at the same time minimize room for errors by assisting human in making decisions. In addition, most of the disadvantages of the traditional solution of preventing tailgating/piggybacking violation can be eliminated by using video analytic technology. The proposed tailgating/piggybacking detection algorithm will be developed on a Linux platform with an external image processing library.

A sophisticated video analytics based system may incur high start-up and operating cost as mentioned in previous section. To minimize the cost of this detection system, the developed algorithm will be implemented on an embedded platform.

While embedded system is significantly cheaper than a desktop computer based system, its resources such as processing power and memory is limited. To ensure smooth real time analysis, improvement and optimization will also be made to the developed algorithm.

1.4 Dissertation Outline

The following chapters for this dissertation are organised as follows:

Chapter 2 discuss the evolution of video surveillance from analogue recording to digital system. The technology of video analytics together with its advantages and limitations is explained. Video analytics techniques that are commonly found in video surveillance such as moving object detection, object classification and tracking are also discussed. An example of video analytics based application is described in the last section of chapter 2.

Chapter 3 describes the overview of this research. This chapter starts with a detailed explanation on the system setup followed by the reason behind the positioning of the camera. System constrains and basic operation of this system is also explained. Finally, the main equipment and also the image processing library used in this research are discussed.

Chapter 4 explains in detail the detection algorithm developed in this research. First, the main modules of the algorithm which are video feed acquisition, moving object detection, connected component analysis and tailgating/piggybacking detection are discussed. A comparison between an advance background modelling method and connected component analysis is made. This is followed by an explanation on how the developed algorithm is optimized. This chapter also includes the explanation of motion templates based algorithm which is the algorithm used to compare against the developed background subtraction based algorithm.

Chapter 5 shows the performance of the developed system in various tailgating or piggybacking situations. Results that were recorded includes accuracy rate, total computational time and also average computational time of the developed detection system. The developed detection system is also compared against a motion templates based system and an analysis is made. This chapter also discussed the major limitations found in the developed detection system.

Chapter 6 is the conclusion of this dissertation starting with a summary of the results gathered in this research followed by the advantages of an embedded based detection system. The potential applications of this detection system are discussed. Some ideas to further improve the developed detection system are also discussed.

1.5 Summary

This chapter discussed the usage of access control in physical security and identified the practice of tailgating and piggybacking as one of the main problems in electronic access control. Most existing solutions have difficulties in preventing this security breach therefore a better way to prevent this problem is proposed. A video analytics based tailgating/piggybacking detection security system will be built using a single internet protocol (IP) camera and embedded based control unit.

CHAPTER 2

VIDEO ANALYTICS AND SECURITY

2.1 Introduction

This chapter begins by exploring the evolution of video surveillance system which is one of the main security features used by authority to monitor relevant events at certain places. Modern video surveillance system can be equipped with video analytics technology to assist security operator. The benefits and limitations of this technology will be discussed. In addition, some video analytics techniques that can be commonly found in modern video surveillance system will also be explained.

2.2 Evolution of Video Surveillance

Video surveillance started with closed circuit television (CCTV) monitoring. The first CCTV was installed in Germany in 1942 by Siemens AG, for the purpose of observing rockets launch. The one responsible for the design and the installation of the first CCTV system was Walter Bruch, a German engineer (Dornberger 1954). After some time, CCTVs are installed in public areas by authorities with the purpose of deterring crime. In addition, some business owners in areas that are prone to theft also follow suit to use video surveillance to improve security of their properties.

Traditionally, analogue cameras in CCTV network are connected by coaxial cables to video monitors. All the videos are recorded on cassette by a video tape recorded for archiving purposes. One of the drawbacks of analogue recording is that quality of video recorded on cassette is inferior compared to digital recording and the cassette needs to be changed every few days due to limited storage capacity (Axis Communications 2012). However, with the advent of digital multiplexer, there is significant advancement in video surveillance. This device enables video feed from several cameras to be recorded at the same time and also added some features that is now deemed standard including motion only recording, which reduced the space needed to stored video.

Digital video surveillance technology has progressed rapidly along with the computer revolution as the cost of digital recording fell. Instead of needing to change tapes every few days, user could record longer duration of surveillance video on hard drive because of video compression and low cost. Digitally recorded video has better quality compared to the often grainy image of analogue recording and it does not deteriorate over time. With digital technology, various enhancements can be carried out to improve the image such as zooming, adjusting brightness and contrast.

The next advancement in video surveillance is linked to the emergence of internet which allows remote access to video surveillance system from virtually anywhere, at any time. Surveillance can be achieved from either from a control centre or a cell phone through internet or local area network (LAN). This is possible because IP surveillance cameras are able to connect directly to the internet. This rise in IP video surveillance is helped by processor advancement, affordable storage cost and better video compression algorithms (Gouaillier and Fleurant 2009). Video surveillance in IP network has several advantages. Its infrastructure is far more flexible than analogue video, IP camera can be connected either by wired (LAN cable) or wirelessly (Wi-Fi). Moreover, any number of cameras can be added to an IP surveillance network system as long the system supports it. Unlike analogue system which is proprietary, IP video surveillance networks use an open architecture which makes it possible to combine hardware from different manufacturers in one security system. In addition, video analytics can be added to IP network video surveillance system to improve security.

2.3 Video Analytics

Video analytics, sometimes also known as "video content analysis" or "Intelligent Video Surveillance" is an active research topic where computer vision is used to perform different tasks by analysing the video feed (Xu 2007). It is used to identify specific object or action in a dynamic scene and ultimately attempts to understand and describe the object behaviour. Video analytics has a wide range of potential applications generally involving the surveillance of vehicle or people such as traffic monitoring in expressway or human detection for security purposes.

Video analytics is getting rapid recognition especially in homeland security in United States. For example, a New York Police Department commissioner has mentioned in an interview that, “A significant part of the video surveillance program going forward will be video analytics, computer algorithms written to automatically alert officers to possible terror attacks or criminal activities” (Stonington and Gardiner 2010). This is a clear indication that video analytics will be one of the key elements in modern video surveillance.

2.3.1 Benefits of Video Analytics

Traditionally, video surveillance is mainly used for post investigation due to some of the limitations posed. One of the well-known problems in security applications is operator fatigue. Various studies show that the ability of a person to focus on a surveillance screen drop by 90% after 20 minutes. A person is also unable to concentrate on 9 to 12 cameras for more than 15 minutes. It has been cited that the ratio between the number of screens and the number of cameras can be between 1:4 and 1:78 in certain surveillance networks. The probability of the security personnel responding immediately to an event captured by a surveillance camera is estimated at 1 out of 1000 which is totally ineffective (Mackworth 1950; Ware et al., 1964; Tickner and Poulton 1973; Green 1999). This is where video analytics can be useful, it can be used to assist human in decision making and cutting down human errors. With video analytics, security personnel can focus their attention only when there is warning issued by the security system and therefore relieves them from monitoring the screen continuously. For example, a video analytics based security system can send a warning to the security control room if there is movement detected in secured places after working hours; security personnel can then take necessary action depending on the situation.

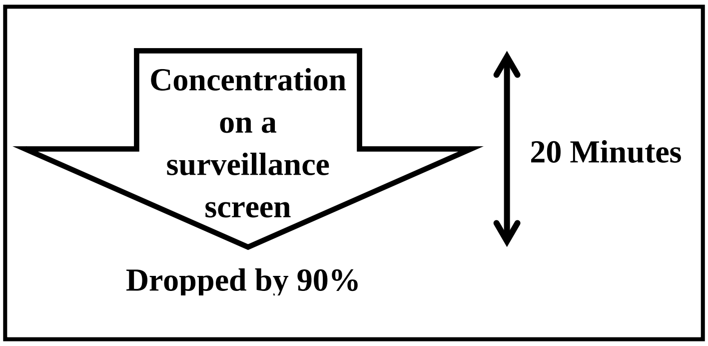


Figure 2.1: Concentration on a surveillance screen of a person dropped by 90% after 20 minutes

Figure 2.2: Security personnel is unable to concentrate on large number of video surveillance screens for a long time (Boymond 2009)

Video of a surveillance area is usually recorded non-stop and a lot time would be needed to properly analyse all the recordings. Instead of spending most of the time observing eventless recordings, video analytics can be used to search for relevant events in the recorded video footage. For example, the full recording can be reduced to parts where only motion is detected which will speed up review process.

In addition, video analytics can operate continuously and expenditure on human resource will also be reduced significantly since fewer operators are needed to monitor the screen. It is also possible to save on operation cost by transmitting or recording only relevant event thus reducing bandwidth and space needed.

2.3.2 Limitations of Video Analytics

Video analytics in real world applications is still a technology with many technical limits especially when analysing complex event (Regazzoni et al., 2010). It is extremely difficult for a machine to distinguish between different human behaviours. For example, a machine would not be able to differentiate between a criminal running to escape from the authorities or a person running to catch a bus.

In addition, there is always a trade-off between the recognition rate obtained and the number of false alarm. Ideally, a security system should have high recognition rate and low number of false alarms. However in reality, a lower detection threshold would result in a higher accuracy rate but at the same time this also raises the potential for false alarm. It is important that a balance must be achieved between recognition rate and false alarm to reduce loss of time and to ensure productivity.

In a nutshell, there is still no perfect system as video analytics can only work in a designated area with certain limitations. Video analytics based security system is usually more effective if deployed in area with few changes; where else a human monitor is more suitable for very active scene.

2.4 Video Analytics Techniques in Video Surveillance

Human operator is proven to be ineffective in monitoring the surveillance screen for a long period of time due to fatigue. Therefore, video analytics are implemented in modern surveillance system to reduce human fatigue and improve security. Video analytics techniques that are commonly found in modern surveillance system are moving object detection, object classification and object tracking. These three techniques form the basis of various video analytics applications such as virtual fencing, human counting and left luggage detection. The following subsections will discuss all of these techniques.

2.4.1 Moving Object Detection

In almost every visual surveillance system, the first step would be detecting movement in the video footage. The method used to identify moving object in video analytics is usually based on detecting changes in a scene. However, detecting changes in video footage does not guarantee the detection of moving object as changes in video scene might be caused by environmental changes. This is a major problem in video analytics because there are many sudden variations in a dynamic scene such as change in lighting (shadows, changes of weather or light reflected by objects) or movement that are not relevant such as the movement of tree leaves and branches. Several moving object detection techniques that are commonly used will be described in this section.

2.4.1.1 Background Subtraction

In many video surveillance applications, background subtraction is one of the most common techniques used to segment out objects of interest in a scene (Stauffer and Grimson 1999; Heikkila and Pietikainen 2006; Maddalena and Petrosino 2008; Pal et al., 2010). This method involves subtracting a target frame with a fixed reference frame. If a pixel value after subtraction is more than the preset threshold, that pixel is considered as a part of the moving object. Background subtraction is easy to implement and it is able to obtain complete object information.

The first step in background subtraction is basic image subtraction.

$$g(x,y) = |f(x,y) - h(x,y)|$$

Let $g(x,y)$ represents the difference between current frame, $f(x,y)$ and reference frame, $h(x,y)$. The result of the subtraction will be converted to absolute value. The last step in background subtraction is to apply thresholding to the difference image, $g(x,y)$.

$$BS(x,y)\begin{cases}0, Background & g(x,y) < T\\ 1, Foreground & otherwise\end{cases}$$

T represents the user preset threshold, it is usually chosen manually by the user depending on the surveillance environment. If the difference is less than the present threshold, result of background subtraction, $BS(x,y)$ will be 0. If it is greater than the threshold level, it is considered as a foreground pixel (Gonzales and Woods 2002).

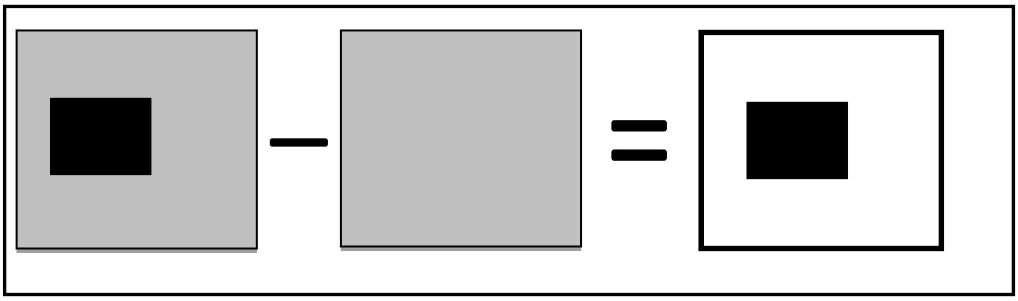


Figure 2.3: Example of background subtraction. Complete information of moving object can be extracted.

The weakness of background subtraction is that it is very sensitive to lighting condition in the scene and it is unable to cope with dynamic background changes such as movement of tree branches, waving leaves and shadows. Therefore, a good background model is important to improve this method effectiveness in detecting moving object (Hu et al., 2004). A codebook based background modelling was proposed by Kim et al. (2005) to handle dynamic background and illumination changes. In their work, the authors quantized sample background values at each pixel into codebooks which represent a compressed form of background model for a long image sequence. With this method, structural background motion over a long period of time can be captured using limited memory.

In another work, Kim et al. (2002) proposed an adaptive background estimation algorithm to cope with the gradual change of illumination. Under this algorithm, background image will be updated by averaging three images including the previous background image if there is no moving object present. The authors also solved the problem of sudden large change of illumination in the background by compensating the average intensity level of the illumination through calculating the intensity difference between the current and background image.

2.4.1.2 Temporal Differencing

Temporal differencing or also known as frame differencing detects regions which have changed through the comparison of video frames separated by a constant time (Lipton et al., 1998). This method is similar to background subtraction but instead of subtracting a fixed reference frame, the current frame will be subtracted with previous frame.

Assume that I_n is the current image and I_{n-1} is the previous image, then the absolute difference between the two image will be

$$\Delta_n = |I_n - I_{n-1}|$$

The difference image can then be thresholded using the same method used in background subtraction

$$TD(x,y)\begin{cases} 0, Background & \Delta_n\,(x,y) < T \\ 1, Foreground & otherwise \end{cases}$$

This method has the advantage of strong adaptability to a variety of dynamic environments but it is not effective in obtaining the complete outline of moving object because holes are often produced (Figure 2.4) in the object (Zhang and Liang 2010). This method also tends to omit some object in the scene especially if it moves slowly.

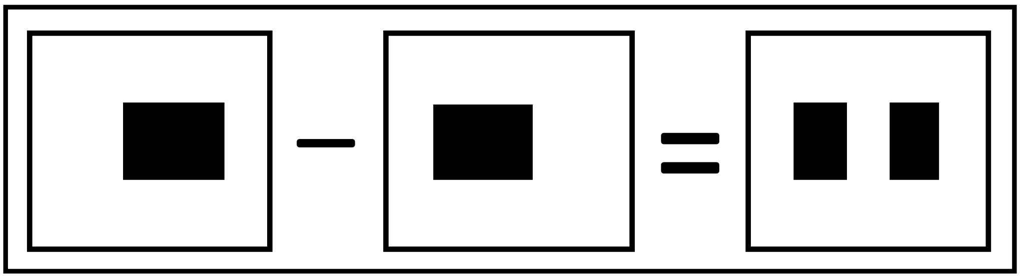


Figure 2.4: Example of temporal differencing. Holes are often produced in the moving object.

There are researches that have been carried out to improve the result of temporal differencing. To improve on processing time, Murali and Girisha (2009) increase the frame difference gap to three frames instead of differencing between current and previous frame. The authors choose to increase the frame gap because in their own experiment using Performance Evaluation of Tracking and Surveillance (PETS) data, it is found that motion of the object between one frame differences is almost negligible, where else unnecessary cast shadow will be generated by fast moving object if the gap is increase beyond three frames.

Temporal differencing tends to include unwanted background caused by the “trailing” object. Lipton et al. (1998) used the knowledge of the target’s motion to crop these unwanted trailing region. The authors achieved this by calculating the difference between the centroid of previous template and the centroid of new template. The region trailing the template is assumed as background material and is cropped so that the new template contains mostly target pixels.

2.4.1.3 Optical Flow

Optical flow based methods can detect consistent directions of pixel change associated with the movement of objects in the scene and can be used to detect moving object between frames without prior knowledge of the content in those frames. For example, Meyer et al. (1998) utilize the information on the optical flow to initialize the contour based tracking algorithm in their research to extract articulated objects which will be used for gait analysis.

There are a lot of optical flow based methods that are available, Barron et al. (1992) evaluated nine different types of optical flow algorithms and found that Lucas-Kanade algorithm is the most accurate and also the least computationally intensive. Lucas-Kanade algorithm assumes that the flow (movement of object) between two consecutive frame is little and almost constant in the neighbourhood of point under consideration. This solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion (Lukas and Kanade 1981).

Although optical flow based algorithm offers better performance of detecting complete movement of an object, most of them are computationally intensive therefore making it hard to implement in real time processing without the aid of special hardware device (Hu et al., 2004).

2.4.2 Object Classification

Typically, once foreground is segmented out from the background by visual surveillance system, it usually contains different types of moving objects. For example, a camera mounted at outdoor would record down moving objects such as cars, human and animals. Therefore it is important to classify them into different categories before further analysis can be done on the objects of interest. Most visual surveillance system will attempt to identify and separate different moving objects into three main categories which are human, vehicle and animals. It should be noted that different classification methods can be combined together to create a classification system with better accuracy and robustness (Jaimes and Chang 2000). The following sections will describe some of the popular object classification techniques used in video surveillance.

2.4.2.1 Shaped Based Classification

One of the main classification techniques is by differentiating objects based on shape. Lipton et al. (1998) used the dispersedness of an object as a classification metric. The main motivation in using this method is because sizes of humans are usually smaller and more complex than vehicles. The Dispersedness of an object is given by:

$$\text{Dispersedness} = \frac{\text{Perimeter}^2}{\text{Area}}$$

Human which has a more complex shape than vehicle will have a larger dispersedness. Lipton et al. also used Mahalanobis distance-based segmentation which provides a better segmentation for classification purpose.

Generally, human have greater height than width while vehicle is wide and short. With this knowledge, Lin et al. (2007) mounted a surveillance camera at street level and use height/width ratio to differentiate between human and vehicle. The reason is that vehicle such as car and lorry usually have a smaller height/width ratio compared to human. This method is also used by the authors to further distinguish between car and motorcycle as car will have a ratio smaller than motorcycle.

2.4.2.2 Motion Based Classification

Another method to classify objects is based on the motion of moving objects. This classification method relies on the knowledge that each object will produce different motion. Bogomolov et al. (2003) used the motion and appearance of a moving object to classify them into vehicle, animal and human. The system developed by the authors utilized Canny edge detector (Canny 1986) to extract motion features from target contour. The authors are able to find eight features to describe the temporal characteristic of motion created by different objects.

Lin et al. (2007) differentiate vehicle and human based on the fact that a moving vehicle will have a constant width but a walking human's width will change periodically due to the swing motion of the legs. The authors applied Fourier transform on the function of object width (as a function of time) to compute the corresponding power spectrum, and then used it to distinguish vehicles and motorcycles from pedestrians.

2.4.3 Object Tracking

After the process of moving object detection and classification, surveillance system generally tracks the movement of object of interested when it appears in the surveillance area. This process requires the system to locate the same object from one frame to another.

Among the notable work in this field is Wren et al. (1997) work. In this work, "pfinder" which is a real time system in tracking people and interpreting their behaviour is successfully built. The developed system tracks a human body by dividing different parts of body such as head, hands and feet into small blobs. The system developed will then slowly build up the model of a person with these small blobs driven by the colour distribution of a person's body. By tracking each small blob, a complete moving human is successfully tracked. The authors have demonstrated the ability of the system by using it in sign language recognition and also gesture control for applications. A few main limitations present in the "pfinder" is that the system is unable to cope with dynamic changes and also the system can only track one person at one time. In addition, Haritaoglu et al. (2000) developed a real time surveillance system, “W4”, for detecting and tracking people in outdoor environment. The system developed does not rely on colour cues and can operate with grey scale video or video from an infrared camera. The authors developed an algorithm that used the combination of shape analysis and tracking to create model of people appearance. The object appearances are modelled by the edge information obtained inside the object silhouette. The limitation of this system is that it is unable to track people correctly when there is occlusion.

2.5 Applications of Video Analytics - People Counting

By combining several video analytics techniques as described in section 2.4, video analytics based applications can be developed. One of the popular applications of video analytics is measuring the traffic of people using camera.

Traditionally, automated people counting is achieved by installing device such as turnstile and rotary bar. These methods suffers from the same problem which is it can only allow one person passes through at a time to ensure accurate counting. By using video analytics, people counting can be done by analysing the images from the video camera. An example of people counting using video analytics is the work done by Albiol et al. (2001). The authors mounted a camera on the top of the train door to count the number of people going in and out of the train carriage. The developed system is able to deal with high densities of people which are usually found at train station. In addition, Chen et al. (2006) proposed a bi-directional counter used to count people flow going through a gate or a door by using area and colour analysis. The authors employed a two stage counting strategy; first the amount of people is estimated using area of people segmented from the background, secondly colour vector extracted from HIS histogram analysis is extracted to refine the initial count. Another research in people counting application which is worthy to be mentioned is by Velipasalar et al. (2006). In this work, the authors proposed an automatic people counting system which is able to calculate people passing through the surveillance area even when they are interacting (merge/split, shaking hands, hugging). The developed counting system learned the person-size bounds which is the interval for size of a single person automatically. The system will calculate the number of people by checking the size of the foreground blob.

People counting can also be implemented in tailgating and piggybacking detection. To detect violation, warning can be issued by the system once the people count is more than one as there should be only one person entering the surveillance area for each access credential presented. The detail explanation to implement this into the detection system will be presented in section 4.7.1.

2.6 Summary

This chapter has presented the role of video analytics technology in modern video surveillance system. Various video analytics technique from moving object detection to object tracking is discussed. There are still many obstacles in perfecting video analytics technology in surveillance system. An inconvenient truth that will always remain is that there will be no perfect video analytics based system as there will always be false alarm. One of the biggest challenges for video analytics based system is to minimize false alarm rate and to handle those false alarm effectively. The deployment of video analytics based security system should not be treated as a perfect security measure. Instead, human operator should always have a thorough understanding on the limitations and capabilities of a video analytics based system and use the system as an aid rather than completely relying on it.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will provide an overview of the detection system developed. To conduct this research, a setup resembling a real surveillance system is built. The proposed and actual setup is discussed in this chapter. The positioning of the camera is one of the crucial elements in building a successful detection system. The reason for installing the camera overhead facing downwards in this detection system will be explained in section 3.2.1. The main equipment used to complete this research which includes an IP camera and an embedded based control unit are described in this chapter.

3.2 System Setup

Figure 3.1 shows the proposed system setup for this project. An IP camera will be installed overhead facing downwards. The IP camera will be connected to an embedded based control unit. The reason for selecting IP camera as the main surveillance camera will be explained in section 3.5.1. The surveillance area will be divided into region A and region B by a single virtual line. Region A is set as the entry region and region B is set as the exit region.

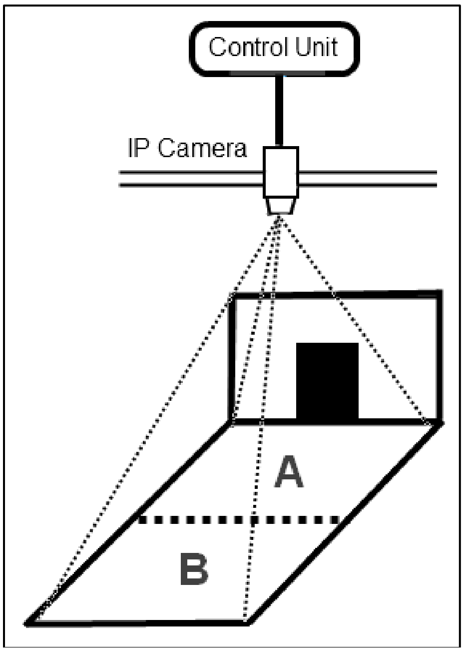


Figure 3.1: Proposed system setup

The height of the camera affects the size of the surveillance area directly. The higher the camera is located, the bigger the field of view of the camera will be and this will result in a larger surveillance area. Larger surveillance area could result in a better detection rate as the moving object will remain in the surveillance area longer when passing through and this allows the system to analyse more frames containing moving object. However, a balance must be found between the height of camera and the size of surveillance area. High camera height may cause problem during deployment while a small surveillance area due to low camera height is not ideal as it is possible for moving object to just skip through the entire surveillance area easily.

Figure 3.2 is the actual setup for this system. A steel frame with the height of around 2.5m is built. This height is chosen as it is almost similar to the typical height for a door. This will allow the possibility of deploying the developed system at places with or without high ceiling (typical ceiling height is around 3m). At the camera height of 2.5m, the effective surveillance area for the system is around 2.6m x 1.9m.

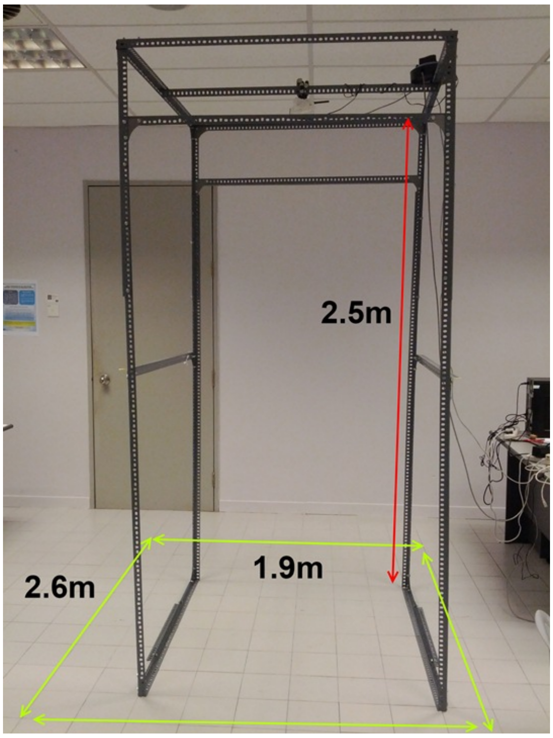


Figure 3.2: Actual system setup

3.2.1 Camera Positioning and the Problem of Occlusion

In most video surveillance systems, the camera is usually installed at an angle less than 45 degrees facing the surveillance area. Cameras that are setup this way faced a problem known as occlusion. Occlusion is a problem where the view of a human is blocked by another human. This is a major issue in implementing video analytics in video surveillance. For instance, in the application of tailgating and piggybacking detection, the system will not able to detect violation accurately if the view of tailgater or piggybacker is obstructed from the camera.

Various researches have shown that the problem of occlusion can be minimized by installing the camera overhead facing downwards. Chen et al. (2006) used a colour video camera installed overhead 4.2m above the floor to count people passing through a door or gate. In the experiment conducted, the authors tested the system by using various people moving patterns such as merging and splitting. By installing the camera facing downwards, the system developed is able to count the number of people that are passing through with accuracy rate of 85% and above in various situations.

It is also observed that Albiol et al. (2001) attached an overhead camera on top of a train door to determine the number of people getting in and out of the train carriage. The movement of crowd in and out of a train especially during peak hour are extremely heavy. By placing the camera on the top of the train door, the problem of occlusion is solved and the system is able to count people accurately with error rate of less than 2%.

There is also a research by Bozzoli et al. (2007) which mounted a commercial low cost camera on the ceiling of a public transport station facing downwards to estimate the number of people passing through the controlled gate. The data collected will allow public transport operator to optimize route allocations and other service. Except from avoiding occlusion, this camera setting also ensures the privacy of passengers is protected by not capturing their faces. The results by the authors showed that the system developed is able to determine the number of people going in and out of a station with accuracy rate of around 95%.

Based on all these different researches, it can be concluded that installing camera overhead facing downwards is one of the easiest and cost effective method to minimize the problem of occlusion. Therefore, this installation method is adopted in this project so that unobstructed view of human walking pass the surveillance area can be captured.

3.3 System Constraints

As with most security system utilizing video analytics technology, there are some important constraints that must be met for the proper functioning of the system. Constraints are set in video analytics based system because it is impossible for a system to handle all different kind of situations that might occur as human behaviour is often unpredictable. Each system can only work at a designated place with specific conditions. The system developed has three main constrains:

1. No one should be inside the surveillance area except if they intend to enter or exit the secured area.
2. Moving objects must only enter the surveillance area from region A and exit from region B or vice versa.
3. The system is designed to handle uni-directional human flows.

3.4 Basic Operation

When deployed, the basic operation of this tailgating/piggybacking detection security system is expected as follows:

1. An authorized person enters the target area by presenting an access credential.
2. The system will start to check for any tailgating or piggybacking violation.
3. Once a violation is detected, the system will alert the security personnel by showing a violation warning on the screen so that appropriate action can be taken.
4. After the door is closed, the system will be reset if there are no people inside the surveillance area.
5. The system can also be reset by the security personnel at any time if needed.

3.5 Equipment

This research is done by using only a few equipment which includes IP camera and also embedded based control unit. The reason for using these equipment and main features of these equipment will be described in this section.

3.5.1 Internet Protocol Camera

IP camera is a type of video camera that can transmit data through a local network or the internet mainly used for surveillance purpose. IP camera is preferred because it has the flexibility to stay connected either wirelessly through Wi-Fi for easy deployment or through LAN cable if a more stable connection is required. With IP camera, the surveillance feed can be remotely accessed and transmission of data will be secured through encryption and authentication methods such as Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access II (WPA2). IP camera is usually able to output video feed in several formats such as H.264, MPEG-4 or Motion JPEG (MJPEG).

The IP camera used in this research is able to support both MPEG-4 and MJPEG format (TP-LINK Technologies 2012). The advantage of MPEG-4 is that this compression method will result in a smaller video size by reducing the quality of images and therefore increasing the amount of video that can be stored. This makes MPEG-4 the preferred format for video archiving. In addition, the small size of MPEG-4 format also reduces the network bandwidth needed for the surveillance system. MPEG-4 encoding gives priority to frame rate when bandwidth available is limited. Image with lower quality will be transmit to ensure the frame rate remain constant. This is not suitable for the developed tailgating/piggybacking detection system as low quality image is harder for the system to analyse.

MJPEG is a video codec where each frame is compressed into an individual JPEG image. This will result in a higher image quality as the compression is independent of the motion in an image. In addition, the latency of processing each image will be lower as each frame is essentially a JPEG image therefore no extra processing would be needed to convert the frame to an editable format. However, the compression level of MJPEG is lower compared to MPEG-4 and will result in a bigger file size for the video. At low bandwidth availability, priority is given to image resolution which means transmitted image would retain the original quality but some frames will be dropped (On-Net Surveillance Systems Inc. 2002). Provided that the dropped frames is minimal, this is an advantage for the developed detection system because receiving fewer high quality frames is better than receiving complete but low quality of video frames which is not suitable for further processing.

In this research, MJPEG is the chosen format as it offers a higher quality images and also lower latency when processing the images. The larger file size of MJPEG video compared to MPEG-4 will not be a concern as the video feed will be processed in real time for the detection of tailgating and piggybacking violation and not for archiving.

3.5.2 Embedded Based Control Unit

A control unit is a device, as its name suggest, used to control the operation of a specific application. In security system, the control unit is usually a desktop computer. However, in recent years embedded system has been steadily gaining popularity in video surveillance applications due to its rapid progress. Currently, embedded based surveillance system can deliver comparable performance compared to a desktop computer based solution with significantly lower startup and operating cost.

There are a few criteria in choosing a suitable embedded based control unit. The embedded system should be small in size as the control unit with a smaller profile will result in an easier installation of the tailgating/piggybacking system. For example, it can be installed into existing settings with minimal modification. The control unit should also feature a processor capable of executing various image processing functions to ensure smooth real time analysis. ARM based processor is a suitable choice in this aspect as it has all the necessary computing capability while maintaining low power consumption at a low cost . The ARM architecture is long known of having the best million instructions per second (MIPS) to Watts ratio as well as best MIPS to cost ratio in the industry. This is proven by the usage of ARM chip in approximately 95% of world’s smartphones (BBC 2011). The control unit should also support open source operating system (OS) such as Linux to lower down the system cost. In addition, the control unit should have all the necessary ports such as Ethernet port and Universal Serial Bus (USB) port. A control unit with open source hardware is also preferred so that modifications to the existing hardware can be done if needed.

Based on all the criteria discussed in this section, the control unit chosen for this research is an ARM based embedded system (BeagleBoard.org

2011) installed with Ubuntu 12.04 (Ubuntu 2012) with XFCE Graphical User Interface.

3.6 Image Processing Library

Due to the lack of a dedicated image processing library in C programming language, a separate library is needed to develop the algorithm. MATLAB and Open Source Computer Vision Library (OpenCV) (Bradski and Kaehler 2008; Bradski 2012) are some of the popular programs used to develop image processing related applications.

```
I = imread('helloworld.jpg');
imshow(I)
```

Figure 3.3: Example of MATLAB code to open and display an image

```
#include "cv.h"
#include "highgui.h"

int main()
{

IplImage* img;
img = cvLoadImage("helloworld.jpg",1);
cvNamedWindow("testwindow", 1);
cvShowImage("testwindow", img);
cvWaitKey(0);
cvDestroyWindow("testwindow");
cvReleaseImage(&img);
return 0;

}
```

Figure 3.4: Example of OpenCV code to open and display an image

MATLAB is a relatively easy language to use as it is a high-level scripting language. For example, a simple program to open and read an image will only takes two lines of code in MATLAB (figure 3.3) but it might takes ten or more lines of code in OpenCV (figure 3.4). However, MATLAB is more computationally intensive therefore more resource is needed to run compared to OpenCV. This is because MATLAB is built on Java while OpenCV is built on C programming language which is closer to machine language code. In addition, MATLAB is a commercial product therefore a license needed to be purchased while OpenCV is an open source library based on Berkeley Software Distribution (BSD) license (Fixational 2012). OpenCV also have higher portability compared to MATLAB which is only supported in Windows, Linux and Mac OS (Mathworks 2013). In comparison, OpenCV is supported across multiple platforms such as Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, Linux and Mac OS.

As cost and speed are the main considerations in this project, OpenCV is the image processing library chosen to develop the tailgating/piggybacking detection algorithm. The OpenCV version used in this research is OpenCV 2.4.1.

3.7 Summary

This chapter discussed the overview of the developed system including the system setup and equipment used to conduct this research. Startup and maintenance cost of a security system are some of the important aspect of a security system. By using an inexpensive IP camera, an affordable embedded based control unit and also utilizing open source library, the cost of the developed system can be kept to an affordable level.

CHAPTER 4

ALGORITHM

4.1 Introduction

The developed algorithm consists of four main modules which are the main focus of this chapter. Flowchart for each of the module will be presented and their function will be explained. The steps taken to optimize the developed algorithm will be discussed in section 4.8. This chapter ends with the introduction to the motion templates algorithm which is also the algorithm used as a comparison to the developed algorithm.

4.2 Main Modules of Detection System

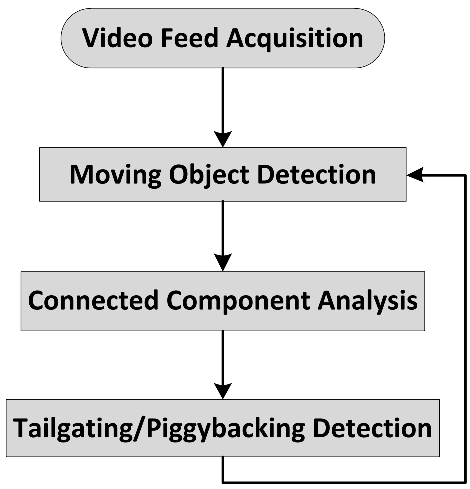


Figure 4.1: Flowchart for main modules of detection system

Figure 4.1 shows the flow chart of the detection system consisting of four main modules. First, the system will attempt to acquire the video feed transmitted by the IP camera. After the video is acquired, the system will proceed to compute the location of moving object with background subtraction technique. The difference image between current and background frame will then undergo connected component analysis so that a clean image consisting of only the moving object can be obtained. The processed image is then ready for the detection of any tailgating or piggybacking violation.

4.3 Video Feed Acquisition

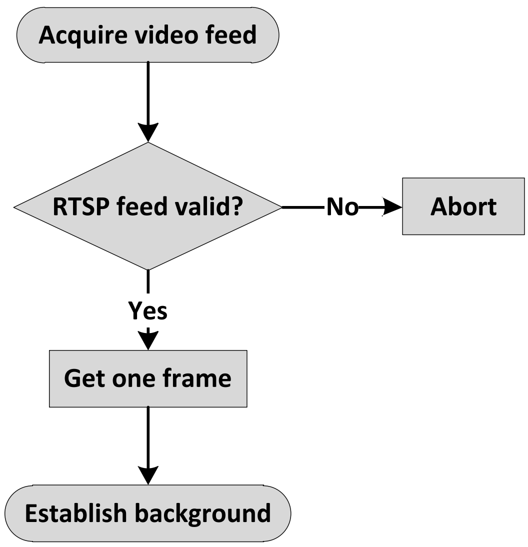


Figure 4.2: Flowchart for video feed acquisition

In this module, the system will first attempt to connect to the video feed from the IP camera. If the real time streaming protocol (RTSP) feed is invalid, this process will be aborted. Once the RTSP feed is validated, the system will get the current frame from the IP camera and establish that frame as the background. The surveillance area should be free from moving object during background establishment so that the background established is an accurate representation of the surveillance area.

4.4 Moving Object Detection

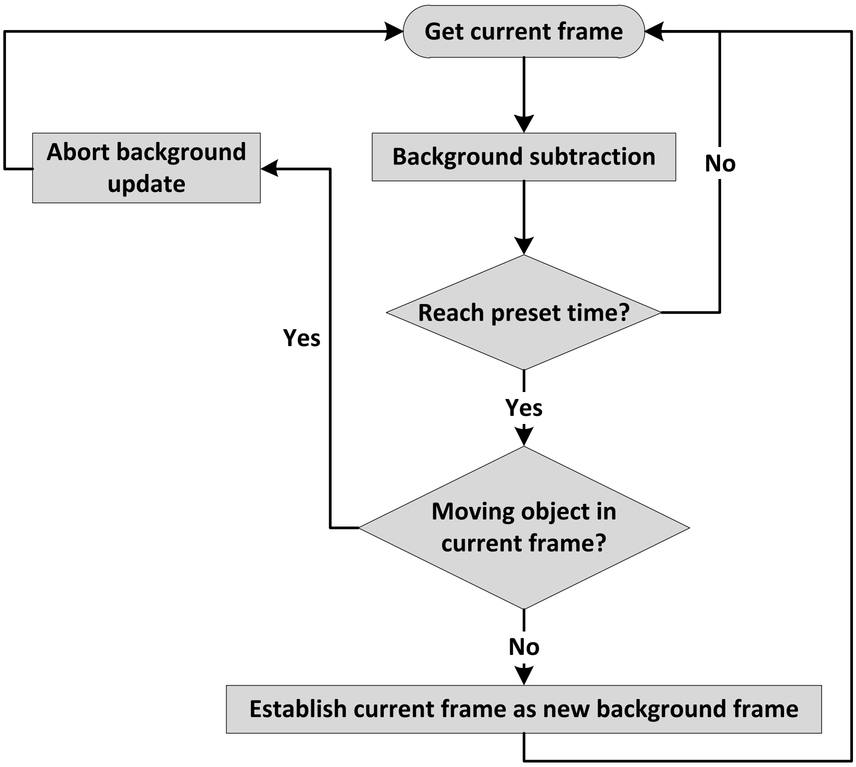


Figure 4.3: Flowchart for moving object detection with background update

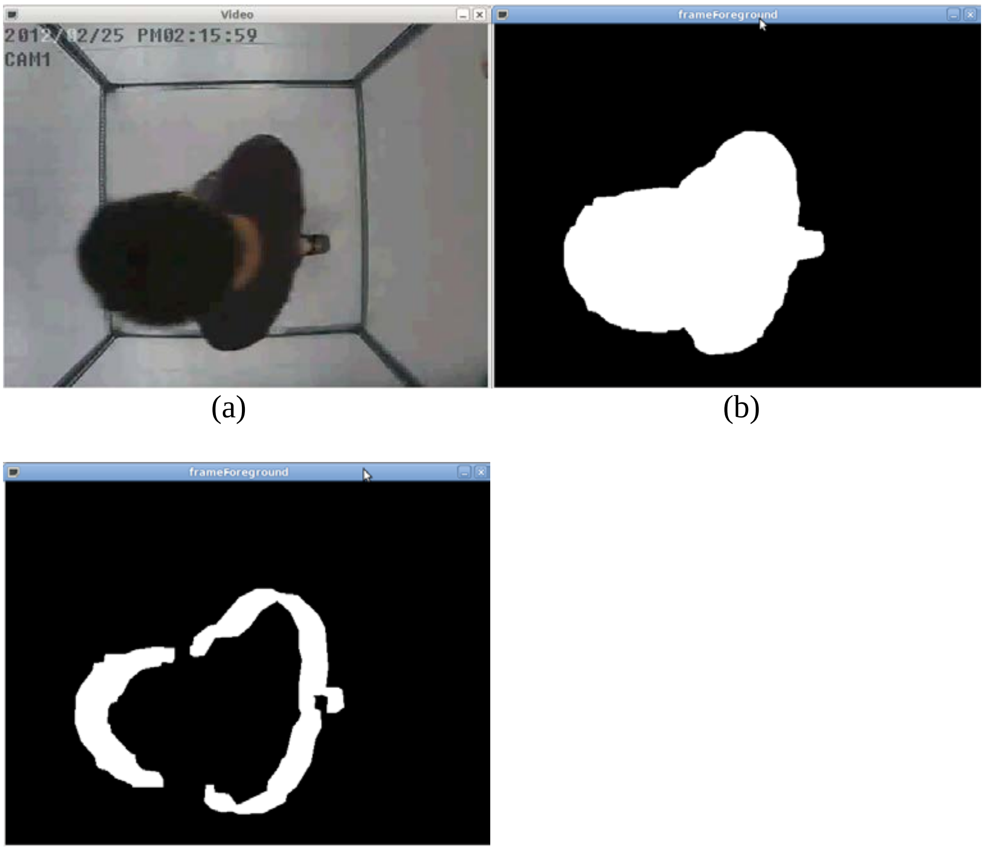
The technique chosen for moving object detection in this system is background subtraction. First, the current frame will be acquired from the RTSP feed. After that, the retrieved current frame will be subtracted with the background frame established in the previous module. By computing the difference between current and background frame, moving object in the surveillance area can be extracted.

As discussed in chapter two, background subtraction is a popular technique used to detect moving object. This technique can be implemented easily and it is able to extract moving object completely. However, background subtraction has a well-known weakness. It is unable to cope with dynamic background. Any changes to the existing background will affect the accuracy of moving object detection. To resolve this shortcoming, real time background update is introduced into the algorithm.

Figure 4.3 shows the flowchart of the moving object detection module with background update. Real time background update is implemented in this system by adding a timer into the algorithm. The timer value set in the developed system is 30 seconds. Once the timer reaches the preset duration, the system will update the background if there is no presence of moving object in the current frame. If moving object is detected, the background update will be aborted and the system will try to establish new background again when the system reaches the preset timer duration. The timer value can be set by the user but it should not be too large; the background needed to be updated frequently as inaccurate background will affect the performance of the system.

4.4.1 Comparison between Background Subtraction and Temporal Differencing

This section will provide a comparison between background subtraction and temporal differencing and the reason for choosing background subtraction as the moving object detection technique in the developed system is explained. Background subtraction and temporal differencing are two techniques that are similar to each other and are commonly used to detect moving object. Each technique has its own advantages and disadvantages as discussed in chapter two. Figure 4.4 shows the difference between both techniques. With background subtraction, a complete contour of the moving object can be extracted as shown in Figure 4.4(b). In temporal differencing technique, the difference between the current and previous image is computed and this will create a “hole” in the moving object as shown on Figure 4.4(c).



(a) (b)

(c)

Figure 4.4: Comparison between background subtraction and temporal differencing. Figure 4.4(a) is the original image; Figure 4.4(b) is the result of background subtraction; Figure 4.4(c) is the result image after temporal differencing

For this research, background subtraction is the more suitable technique as the algorithm required a complete contour of the moving object for accurate detection. The advantage of frame differencing has over background subtraction is the ability to cope with changes in background such as illumination. However, by introducing background update in the original background subtraction algorithm as explained in previous section, background subtraction will have the same ability to cope with these changes.

4.5 Connected Component Analysis

As discussed in chapter 2, background subtraction technique relies on advance background model for better detection of moving object. However, advance background modelling is not used in the developed algorithm so that processing time can be improved as this algorithm will be implemented on an embedded platform which has limited resources. For example, advance background modelling such as background averaging (section 4.6.1) required 4-5 seconds to establish the background with the averaging of 80 frames which is too slow. Due to the lack of an advance background modelling, the thresholded image after background subtraction is not suitable for further analysis. A clean image and a complete contour of the moving object are crucial for an accurate detection of any tailgating or piggybacking violation. Therefore, connected component analysis is introduced in this algorithm.

4.5.1 Basic Theory of Connected Component Analysis

One of the most important tasks in when analysing an image is to determine which part of an object is connected physically. By determining which part is connected together, a complete contour of an object can be extracted from the image. The fundamental concept in connected component analysis is to find the connectivity between pixels. To determine if two pixels are connected, it must first be determined if they are neighbours (Gonzalez et al. 2002).

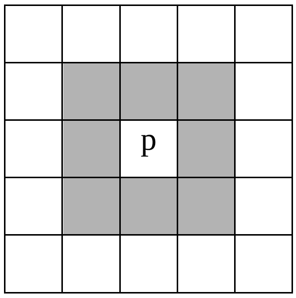


Figure 4.5: Neighbours of a pixel

Figure 4.5 is a small region of an image at pixel level. The neighbour of a pixel is the set of pixels that connected to it. In figure above, there are 8 pixels (in grey colour) that are “touching” the pixel “p” and are considered as neighbour of “p”. A single pixel can have a maximum of 8 neighbours.

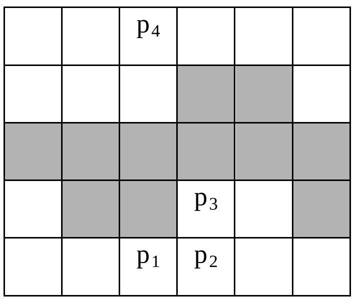


Figure 4.6: Chain of connection between pixels

Two pixels can also be considered as connected even when they are not next to each another provided there is a chain of connection between the two pixels. Referring to Figure 4.6, p_1 is said to be connected to p_3 bacause p_1

is connected to p_2 and p_3 is also connected to p_2. Therefore, all white pixels around p_1, p_2 and p_3 are considered as connected to each other. However, p_4 and the white pixels around it are not connected to any of the pixel p_1, p_2 or p_3 as the grey pixels in the middle of the Figure 4.6 has blocked the connectivity. By applying this concept to a whole image, connected regions of segmented object can be identified.

4.5.2 Implementation of Connected Component Analysis in Developed System

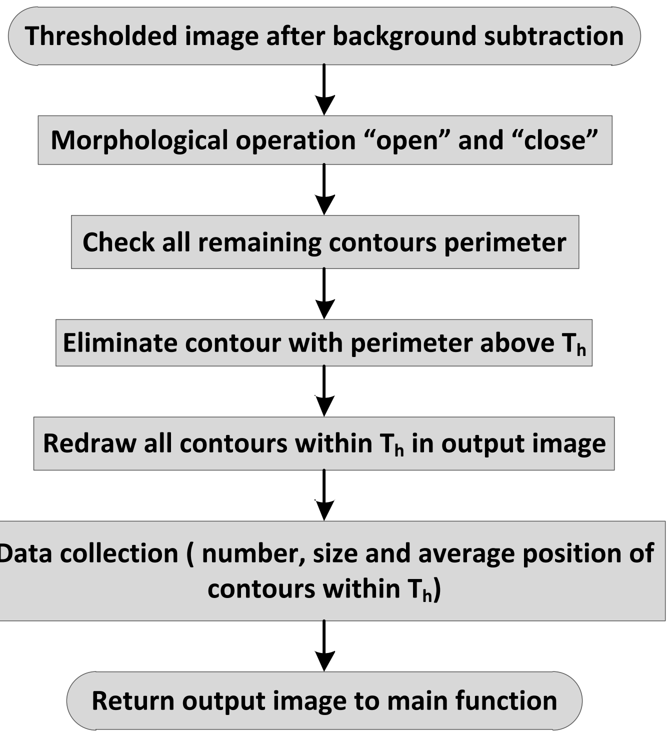


Figure 4.7: Flowchart for connected component analysis

Figure 4.7 is the flowchart for connected component analysis. First, morphological operation "open" (erosion followed by dilation) and morphological operation "close" (dilation followed by erosion) (Bradski and Kaehler 2008) will be applied on Figure 4.9 which is the thresholded image after undergoing background subtraction. This image has a lot of unwanted contours from the background. As seen on Figure 4.10, small random noises around the moving object's contour in Figure 4.9 have been eliminated by morphological operation "open". The surviving components from the previous operation are then rebuilt using morphological operation "close". A complete contour of the moving object is extracted successfully but at the same time some unwanted contours that are supposed to be background are also rebuilt in this process. The next step which is contour filtering is designed to solve this issue. In this process, contour will be filtered out based on their size. Contours with perimeter less than T_{h} will be deemed as too small and will be eliminated as it should belong to background. T_{h} is given by:

$$T_{\mathrm{h}} = \frac{(\mathrm{F_H} + \mathrm{F_W})}{S}$$

$\mathrm{F_H}$ is the frame height and $\mathrm{F_W}$ is the frame width. S is a user preset scale, the smaller the scale, the bigger the contour size needed to be to remain in the foreground. For camera height of 2.5m, it is found through experimental work that the suitable value for S is two.

Figure 4.11 is the final output of the connected component analysis and the complete contour of moving object without any noise is extracted and all unwanted contours successfully eliminated.

Before the output image is return to the main function, some data are collected for the use of tailgating and piggybacking detection. This includes the number of contours within the threshold and also the size and the position (X coordinate) of these contours.

Figure 4.8: Original image

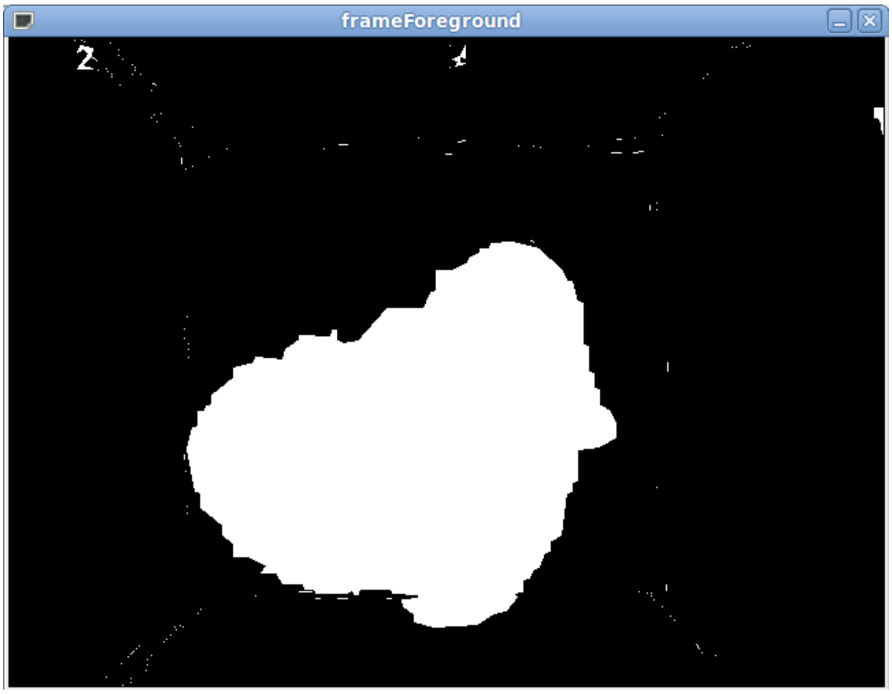


Figure 4.9: Thresholded image after undergoing background subtraction

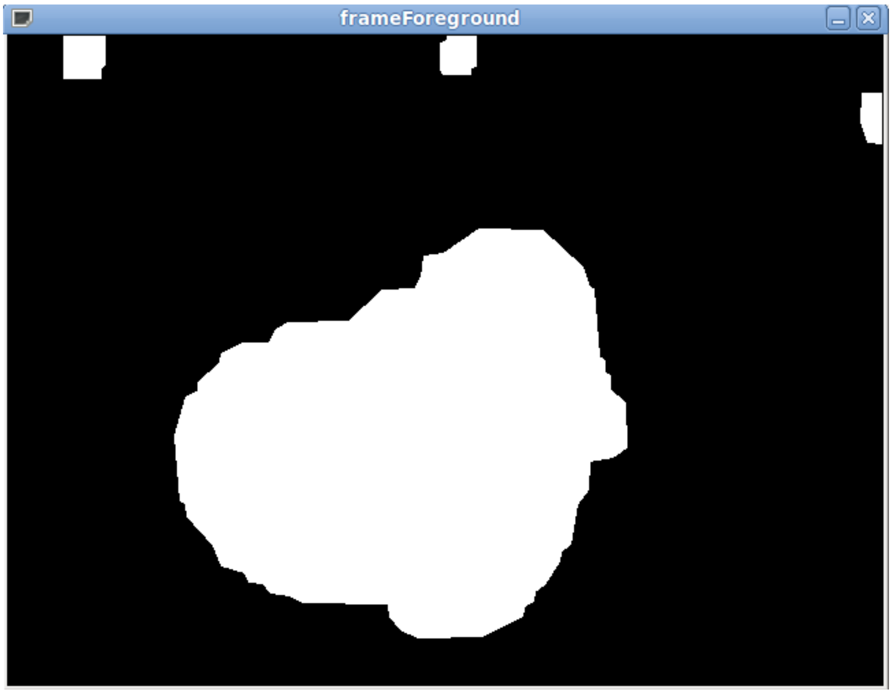


Figure 4.10: Image after undergoing morphological operation “open” and morphological operation “close”

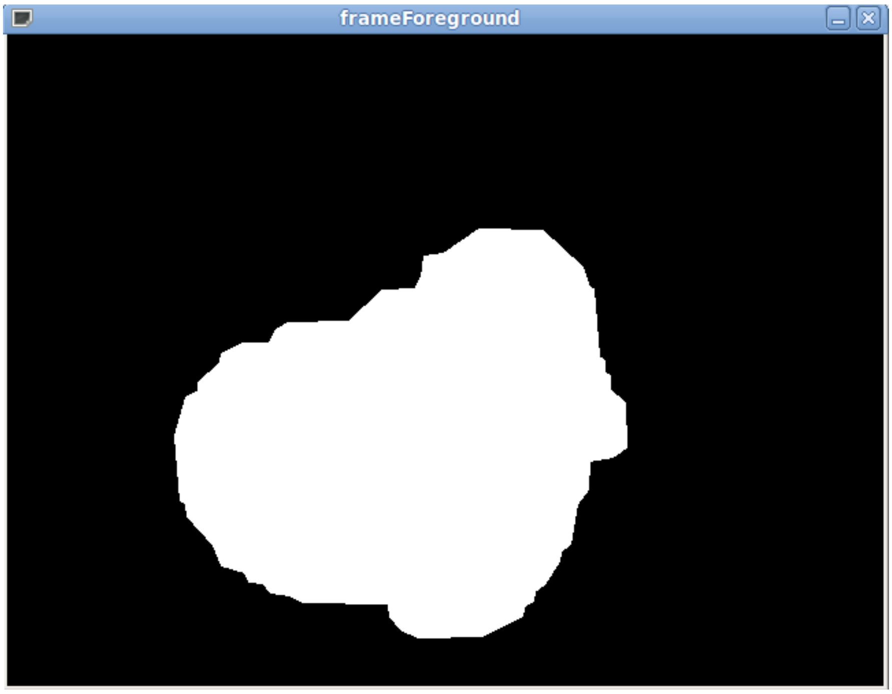


Figure 4.11: Image after completing connected component analysis

4.6 Comparison between Advance Background Modelling and Connected Component Analysis

In this system, connected component analysis is used to replace advance background modelling in background subtraction technique to improve the output image. This section will provide a comparison in using advance background modelling and connected component analysis.

4.6.1 Background Averaging

The background modelling method chosen for comparison is background averaging. In this averaging method, several images of a background scene without moving object are taken from the exact same position. This method assumes that the noise in the image captured is random. This way, random fluctuations above and below actual image data will gradually even out when more and more images are averaged (Cambridge in Colour 2012). In short, background averaging works by adding together a number of images, and then the result is divided by the number of images used. This technique can be found in digital camera where several images are taken in quick succession and then combined together to create a low noise image.

With background averaging, the image can be expressed as

$$I(x,y) = \frac{1}{k}\sum_{i=1}^{k}\big(f(x,y) + n_i(x,y)\big)$$

where f(x,y) is the sum of the images used and n_i is the noise found in the image. The operation is to sum up k images and divide it by k. Theoretically, if there are infinite numbers of images, sum of noise would be zero.

$$\sum_{i=1}^{k} n_i(x,y) = 0$$

Therefore, the result from background averaging is an image with no noise.

$$I(x,y) = \sum_{i=1}^{k} f(x,y)$$

4.6.2 Comparison Result

The original image from Figure 4.8 is used in this comparison test. Figure 4.12 shows the output image if no averaging is applied. This image contains noise and unwanted background artefacts. For comparison, 80 images are used for background averaging and Figure 4.13 shows the result of it. Compared to the original image in Figure 4.12, this output image has improved considerably. Some of the noise around the moving object contour has been eliminated. However when compared to Figure 4.11 which is the output image of connected component analysis, it is observed that connected component analysis performed better as it is able to extract a cleaner image that consists only the moving object's contour from the original image.

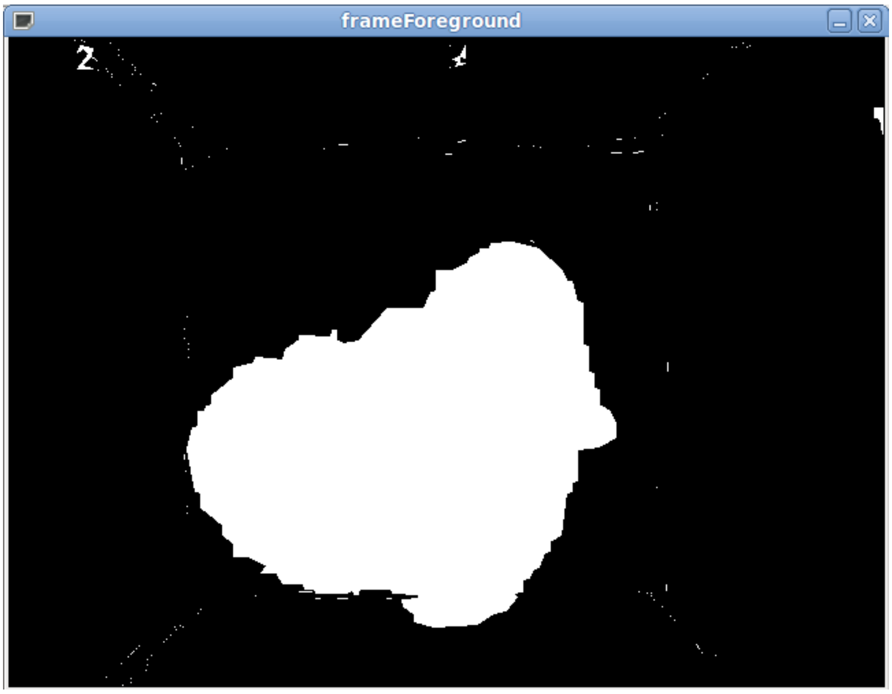


Figure 4.12: Thresholded image with no background averaging

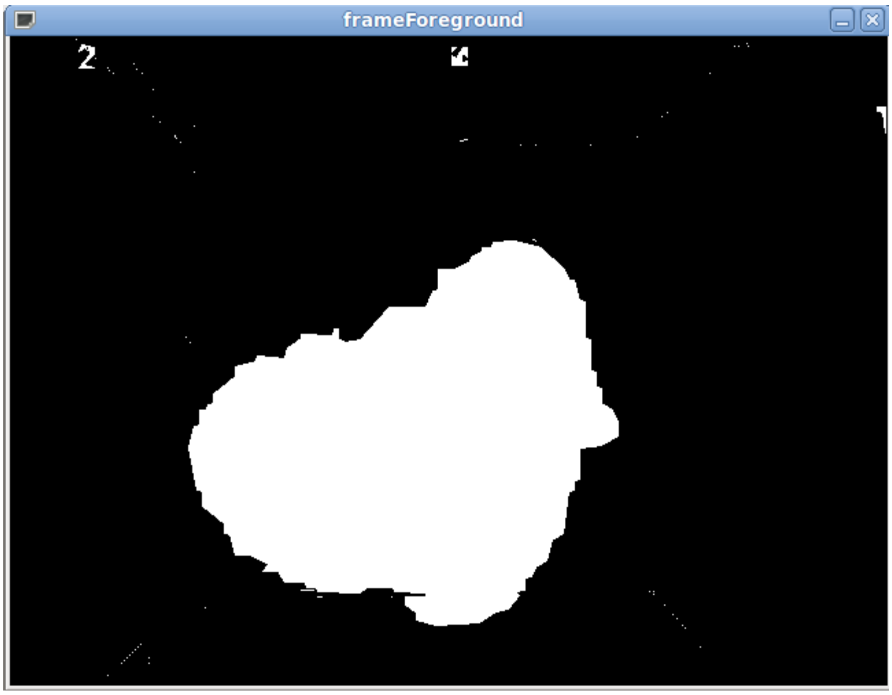


Figure 4.13: Thresholded image with background averaging of 80 frames

Another downside in using background averaging is that background establishment is not instantaneous. Depending on the number of frames being averaged, there will be some delay as multiple images are being added together and then averaged out. The image will be cleaner when the number of images used is larger but more time will be needed for the system to process all those images. For the developed system, establishing a background with 80 frames of images on the embedded system will take 4-5 seconds. Therefore, the surveillance area needed to be empty for a certain time while the background is being established. This will cause inconvenience if there are people that intend to enter the restricted area at that time.

4.7 Tailgating/Piggybacking Detection

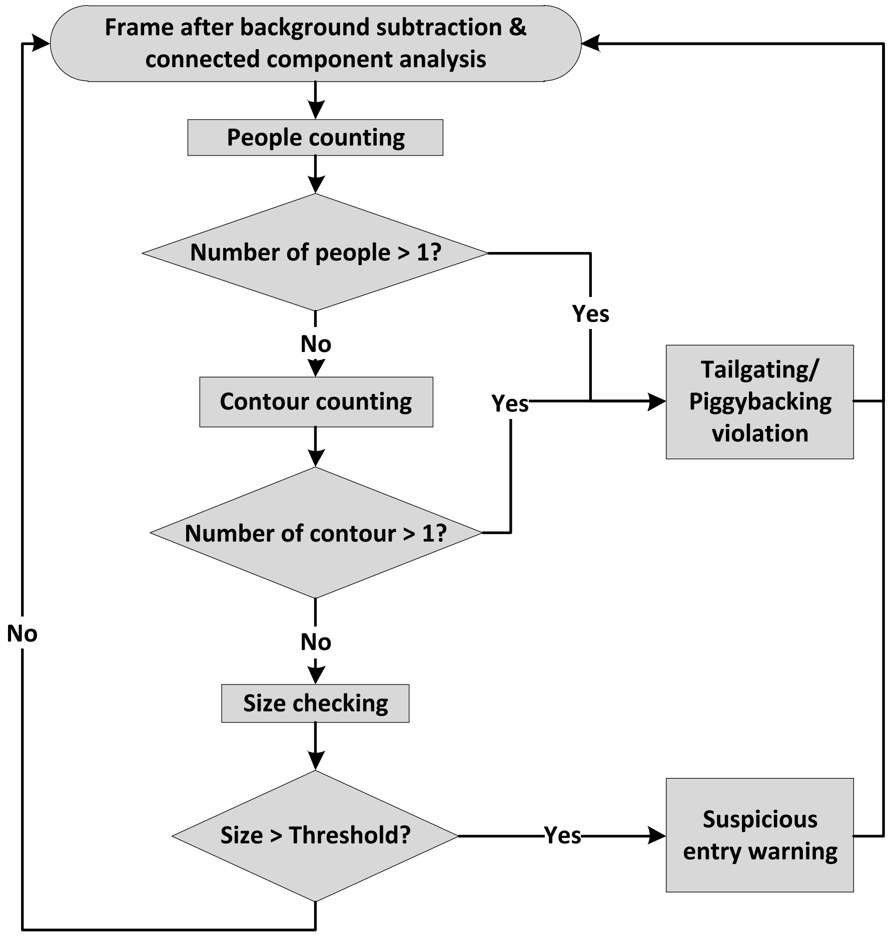


Figure 4.14: Flowchart for tailgating/piggybacking detection

After the process of connected component analysis, the image is now ready for the detection of tailgating or piggybacking violation. Under normal circumstances, there should be only one person entering the secured area if one access credential is presented. Therefore, the key in detecting tailgater or piggybacker is the ability of the security system to detect the second person, whether authorized or not, that passes through the surveillance area.

Figure 4.14 shows the flowchart of tailgating and piggybacking detection module. The developed system employed a three stage checking to detect violation. The first stage is by counting number of people that pass through the surveillance area, second is by checking the number of people inside the surveillance area and the last stage is by checking the size of the person that is in the surveillance area for any suspicious entry activity.

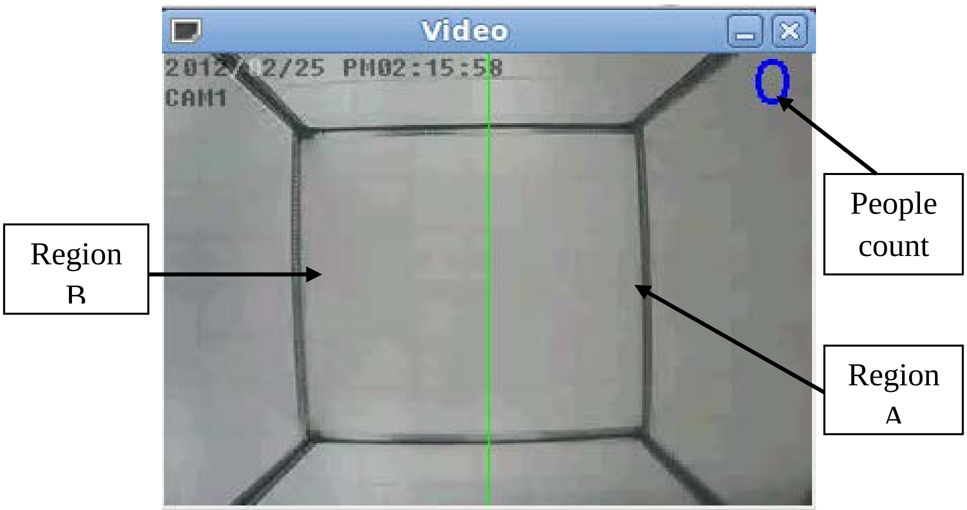


Figure 4.15: Surveillance area

Figure 4.15 shows the surveillance area captured by an IP camera. It is divided into two regions by a single virtual line set almost in the middle.

Region A on the right is the entry region and region B on the left is the exit region. The number shown on the top right of the screen is the people count. Violation and suspicious entry warning will be shown on the bottom of the screen in red colour font.

4.7.1 First Stage: People Counting

There have been a lot of researches in the field of people counting using video camera. Most of the systems are designed to be used in places with high volume of human flow and equipped with the ability to handle complex interaction between human (Albiol et al., 2001; Chen et al., 2006; Hou and Pang 2011). Due to the complexity of these algorithms, a number of them are implemented on desktop computer. However in this research, the resources on the implemented platform are limited. Therefore a simple yet effective algorithm needed to be developed to ensure smooth real time analysis. In addition, advance counting algorithm is not a necessity in anti-tailgating/piggybacking system as it is usually deployed in indoor environment with minimal human flow passing at the same time.

The developed system counts by tracking the position of people passing through the surveillance area. If a person walks through from region A to region B, people count will be increased by one. Tailgating/Piggybacking violation warning will be issued when people count is more than one because there should be only one person passing through each time a credential is presented. People that are exiting the secured area (Region B to Region A) will not be counted.

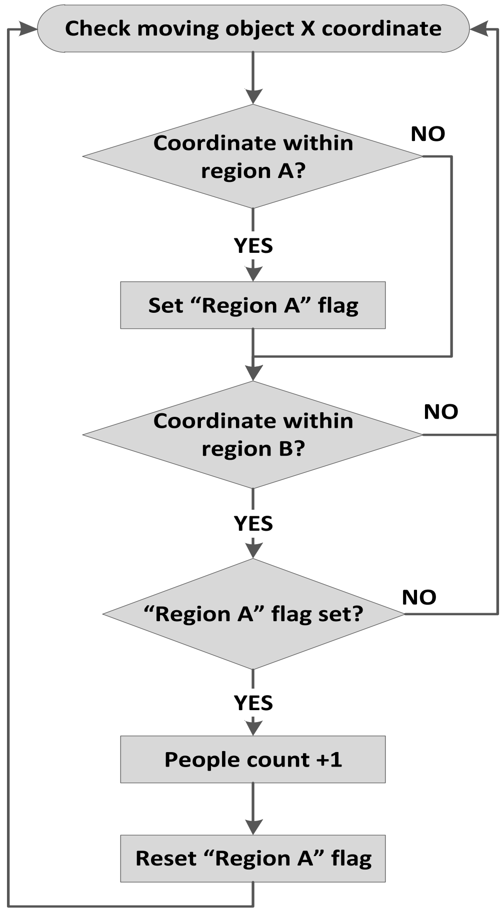


Figure 4.16: Flowchart for people counting

Figure 4.16 is the flowchart for the people counting algorithm. The system will first check the position of the person inside the surveillance area by checking its X coordinate. This information is collected in the connected component analysis module. If the system detected that the person X coordinate is within region A, a flag named as "Region A" will be set. After that, the system will continue to check for the person coordinate. Once the coordinate of the person is within region B, the system will check the "Region

A” flag status, if it is set then the people count will be increased by one and then the “Region A” flag will be reset.

Figure 4.17: Violation warning when people count is above one

4.7.2 Second Stage: Contour Counting

In addition to people counting, the system is also constantly counting the number of people inside the surveillance area. The system does this by counting the amount of contours in the surveillance area. If there is more than one contour in the surveillance area at the same time, this means that there is more than one person in the surveillance area and the system will identify this as a tailgating/piggybacking violation. The logic behind contour counting is that since there should be only one person entering the surveillance area each time an access credential is presented, so there should be only one contour present at the surveillance area. However, the implementation of contour counting has resulted in a system that is only capable of handling uni-directional human flow. If two persons enter and exit the surveillance area at the same time, the contour counting stage will not be able to function correctly. This is a disadvantage but it will improve the violation detection rate. Contour counting is designed to complement the people counting stage. If the people count is wrong due to false positive or false negative, the system is still able to detect violation through contour counting.

Figure 4.18: Violation warning when number of contours is above one

4.7.3 Third Stage: Size Checking

The third and final stage involved the checking of the person size to detect suspicious entry behaviour. Any person passing through the surveillance area with a size that exceeded the preset threshold is deemed as suspicious entry. This stage is a final backup if the previous two stages failed to detect violation. There are several situations that will result in contour with large size. One possible situation is when there are two people walking side by side together in an attempt to avoid detection. In addition, people that attempt to run through or jump over the surveillance area will also result in a large contour size and will trigger the suspicious entry warning.

Figure 4.19: Suspicious entry warning when contour size is above threshold

4.8 Algorithm Optimization

From the result in the section 5.6.1, it is noted that connected component analysis has the highest computational time compared to the other three modules. Connected component analysis is an important part of this algorithm as shown in section 4.5, it is observed that it is harder to analyse the image to detect tailgating or piggybacking violation without connect component analysis. To improve on the system performance, this research proposed to reduce the usage of this module by limiting the area analysed in inactive scene (no moving object detected in the surveillance area).

Figure 4.20 shows the flowchart for algorithm optimization. The system will first check for the presence of moving object in region A by counting the number of contour. If the contour in region A is less than one (no moving object present), the region of interest (ROI) will be set to region A only. This will reduce the area needed to be analysed by connected component analysis module by approximately 50%. If the amount of contours in region A is equals or more than one (moving object present), ROI will be reset so that the whole surveillance area will be analysed.

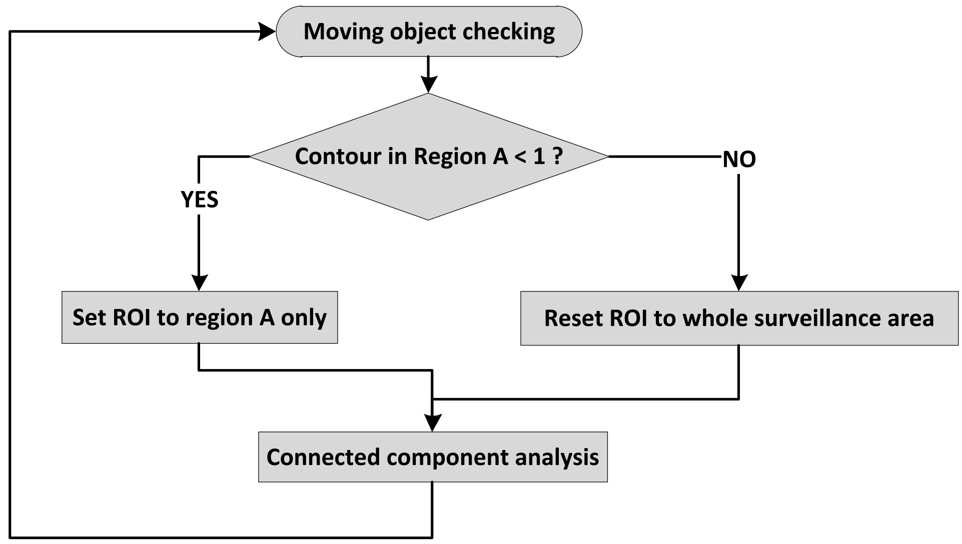


Figure 4.20: Flowchart for algorithm optimization

4.9 Motion Templates Based Algorithm

A separate, motion templates (Davis and Bradski 1999) based algorithm is used as a comparison with the developed background subtraction based algorithm. Motion templates is an effective technique to track general movement of an object and are also useful in gesture recognition applications. Motion templates is chosen because the basis of this algorithm is temporal differencing which is similar to background subtraction used in the developed algorithm. Therefore, motion templates can be implemented into the existing algorithm with minimal modifications. This will ensures that a fair comparison can be made. The results of the comparison will be discussed in the next chapter.

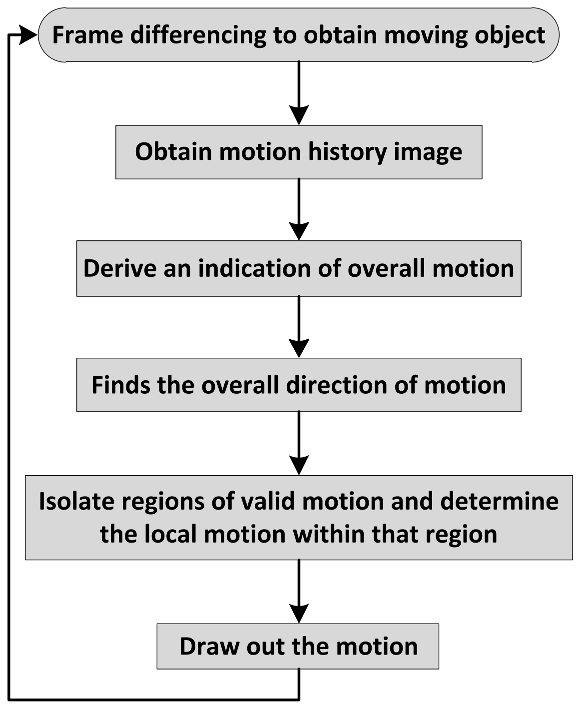


Figure 4.21: Flowchart for motion templates algorithm

Figure 4.21 is the flowchart for the motion templates algorithm. Motion templates technique requires a complete or part of a moving object's silhouette to operate. The technique chosen to obtain the silhouette is by using frame differencing or also known as temporal differencing. Once the object's silhouette is acquired, the system will obtain the motion history image which is the history of the movement of an object. After that, the indication of overall motion is obtained by analysing the gradient of the motion history image. Large and invalid gradient will be eliminated in this process. In the next step, the overall direction of the motion will be determined by summing up the pre-computed motion vectors. Next the system will segment out the regions with valid motion and determine the local motion. In this process, motion history image will first be analysed for any current silhouette. Once a silhouette has been established, the system will go around the perimeter of that silhouette to search for nearby recent silhouette. All these silhouettes will be segmented out and then the local motion will be computed. This process will be repeated until no current silhouette is available. Lastly, the moving objects motion will be drawn out.

4.10 Summary

This chapter has discussed all the four main modules of the detection algorithm. Among all the modules, connected component analysis is the most computationally intensive module but also the most important in this detection system. A successful video analytics system depends heavily on the system's ability to extract the complete contour of the moving object without noise and unwanted contour. With connect component analysis module, the desirable output image can be achieved. The performance of the detection system based on algorithm in this chapter will be discussed in the next chapter.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Introduction

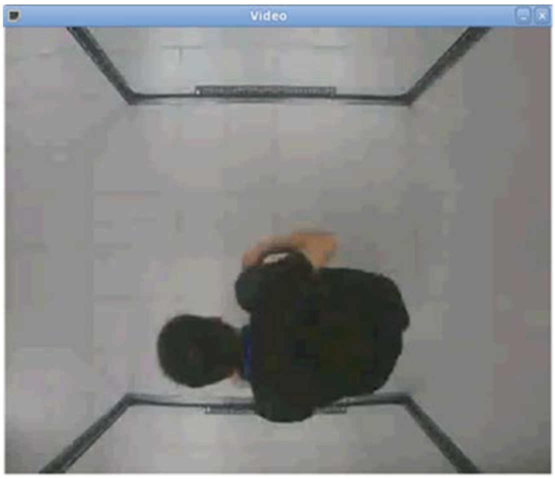
This chapter focuses on the results gathered from different tests with the recorded videos. Videos containing various situations of people entering the surveillance area will be discussed in this chapter. The developed detection system's performance will also be presented here. The parameters used to measure the performance of the detection system are accuracy rate and computational time. To benchmark the developed detection system, the performance of the developed background subtraction based system is compared against the performance of a motion templates based system in section 5.7. This chapter ends with the discussion of the limitations of the developed system.

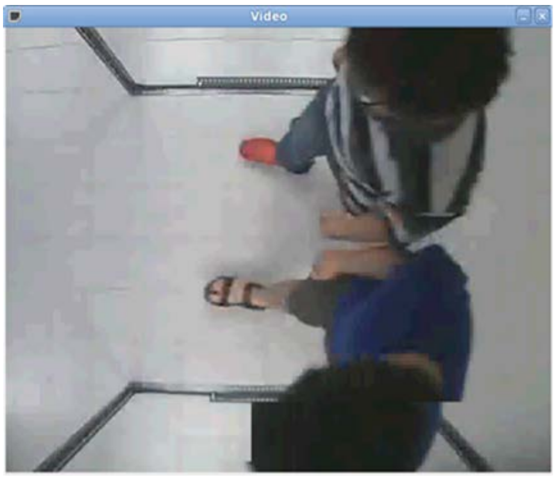
5.2 Recorded Videos

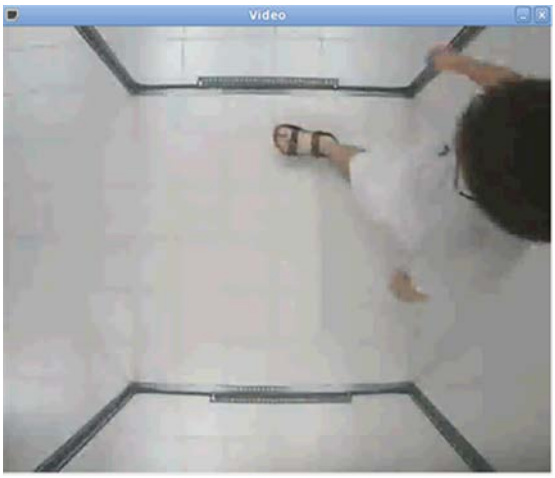
In order to verify the performance of the system, videos simulating various situations of people entering the surveillance area are recorded. The videos recorded contained situations where people enter the surveillance area normally and also situations where people attempt to beat the detection system. Unless stated, all the videos (MJPEG format) are recorded in an indoor office with sufficient lighting (370 Lux) with an IP camera mounted overhead 2.5m from ground. Recorded videos are then processed on the ARM-based embedded system.

Table 5.1: Summary of Recorded Video

Situation	Video Screenshot and Description
Walking	
Running	
Jumping	

Sneak in	• The subject will try to appear as small as possible to the camera when passing through the surveillance area. The methods tested are squatting down and also sneaking in from side  
Carrying/Pushing object	• The subject will pass through the surveillance area by pushing/carrying a trolley/chair/box 

	 
Side by side	• In this video, two persons will pass through the surveillance area by walking side by side with each other to escape detection 
Following closely	• In this video, two persons will pass through the surveillance area together. The person behind will follow the person in front closely

	
Low light	• The only light source is the natural lighting from the office's windows • Lighting condition: 15.4 Lux 
Clothing colour similar with background	• Moving object will wear white colour clothing as it is similar with the colour of the floor (background) 

5.3 System Accuracy

Table 5.2: System Accuracy

Situation	**Accuracy**			
	5 FPS	**7 FPS**	**10 FPS**	**15 FPS**
Walking	100%	100%	100%	100%
Running	50%	100%	100%	100%
Jumping	67%	100%	100%	100%
Sneak in	100%	100%	100%	100%
Carrying/Pushing object	100%	100%	100%	100%
Side by side	50%	50%	50%	50%
Following closely	100%	100%	100%	100%
Low light	67%	67%	100%	100%
Clothing colour similar with background	100%	100%	100%	100%
Average Accuracy Rate	**81.60%**	**90.80%**	**94.40%**	**94.40%**

Table 5.2 shows the accuracy of the developed system for each video for different frames per second (FPS). The accuracy is computed by checking the amount of correct warning issued by the system. However, suspicious entry warning is ignored in this test as it is neither correct nor wrong.

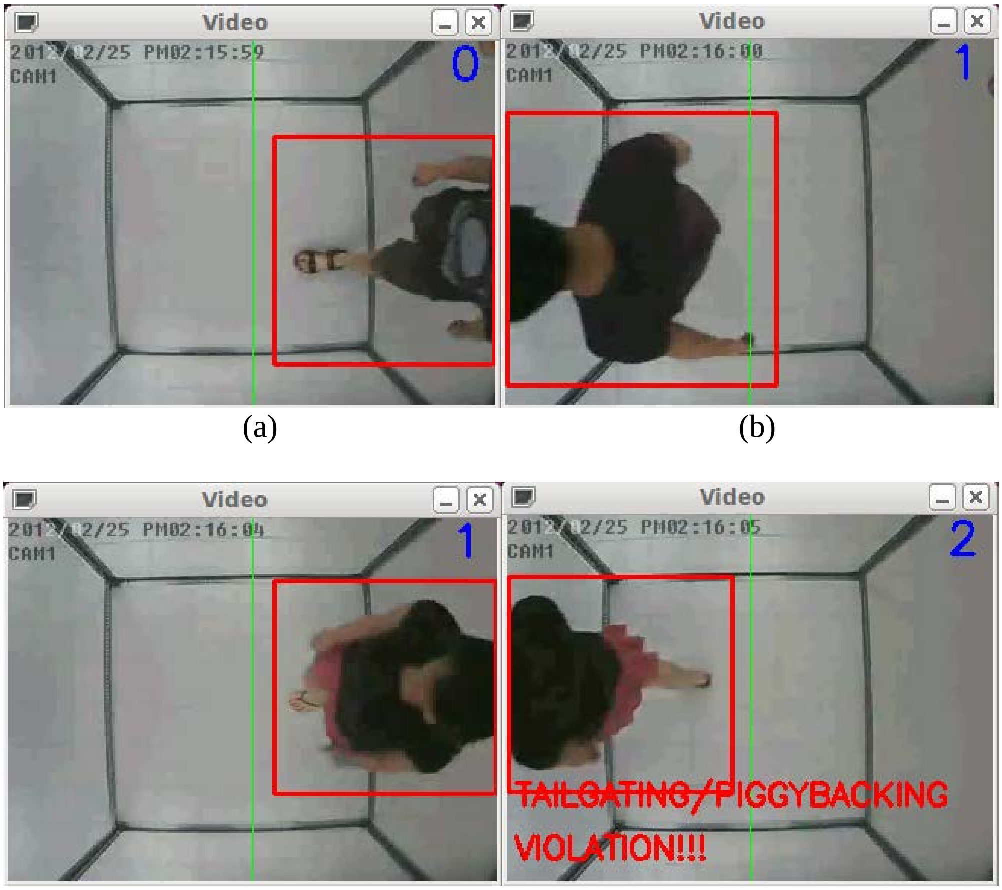
The results show that this system is able to identify tailgater/piggybacker accurately in five situations tested. Perfect accuracy rate of 100% can be achieved in walking and sneaking in situation. This system can also identify violation correctly in situation where a person carry or push an object when passing through the surveillance area. However, if the object carried is too big and caused the moving objects contours to exceed the size threshold, the system will detect it as a false positive and wrongly issued a suspicious entry warning. This system is also able to achieve an accuracy rate of 100% for situation where two people pass through the surveillance area by walking closely together. In this situation, the people count stage actually computed the wrong number of people passing through but tailgating/piggybacking violation is still detected successfully through contour counting. The developed system can also handle situation where the clothing of moving object has a similar colour with the background. In background subtraction technique, moving object that has a colour similar to the background has a high chance of escaping detection as it will blend with the background. From the colour similarity with background video it is found that the colour of hair and skin of a person which has a different colour from the background helped in detecting the moving object.

For video with 5 FPS, this system has difficulty detecting violation in situation where a person run or jump through the surveillance area. These fast moving objects have successfully avoided detection due to the low frame rate used. As explained in the section 4.7.1, this system verify the number of people passing through the surveillance area by checking if the person moves from region A to region B. In running or jumping situation, the person might only appear in one frame which caused the system unable to detect any human passing through because a minimum of two frames (One in region A and one in region B) that contain the moving object are required for the system to count. Although the system is unable to count fast moving object accurately with 5 FPS, there is a high possibility suspicious entry warning will be triggered as fast moving objects usually have big movement which will result in a large contour size. This problem can be solved by increasing the frame rate to 7 FPS where both running and jumping scene can be detected with 100% accuracy.

In the case of two people walking side by side, violation is not detected as this system will detect it as only one person passing through under all different frame rates. This is due to the contours of those two people have merged together as one as they are too close together. However, the system will identify this situation as a suspicious entry due to the large contour size.

A video was also recorded to test this system performance in low light condition. Both video with 5 FPS and 7 FPS only managed to achieve an accuracy rate of 67%. The reason for lower detection rate is similar to the situation with fast moving object where the moving object does not appear in enough frames for it to be counted. This system can only detect the moving objects in some frames due to the bad lighting condition. A minimum of 10 FPS is needed to achieve 100% accuracy.

It is observed that from the accuracy test, the contour counting stage in tailgating/piggybacking detection is proven to be able to complement the people counting algorithm by detecting violation even when the people count is wrong. Size checking stage is also fairly accurate in identifying suspicious entry activity.



(a) (b)

(c) (d)

Figure 5.1: Walking video from one person situation (7FPS). (a) to (d) show that tailgating/piggybacking violation warning is issued by the system once the people count (number on top right of each screenshot) is 2 and above

(a) (b)

Figure 5.2: Sneaking in video (7 FPS). In this video, tailgating/piggybacking violation is detected even when a person is trying to sneak through by walking at the side of the surveillance area

(a) (b)

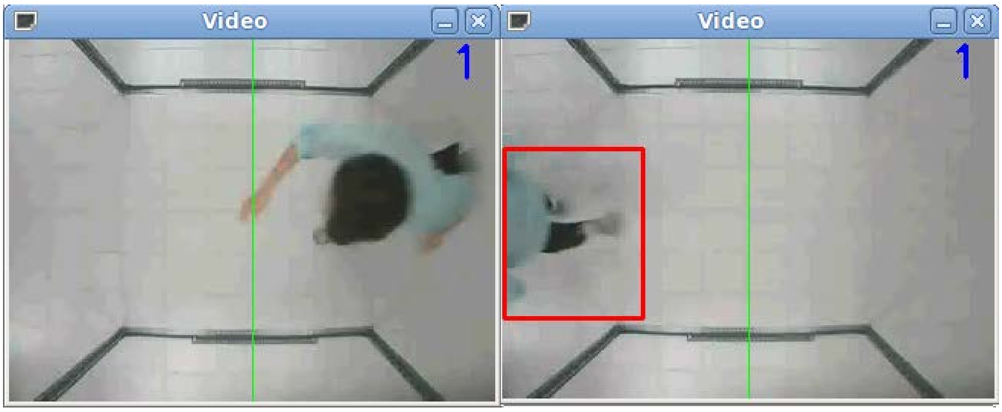
Figure 5.3: Carrying/pushing object video (7 FPS). In (a), suspicious entry warning is wrongly issued by the system. The system will switch the warning to tailgating/piggybacking violation once the people count is more than one

(a) (b)

Figure 5.4: Following closely video (7 FPS). The system will issue a warning when there is more than one person in the surveillance area at a time

(a) (b)

Figure 5.5: Clothing colour similar with background video (7 FPS). Violation can be detected successfully in this situation



(a) (b)

Figure 5.6: Running video (5 FPS). This moving object is only detected in region B as shown in (b) due to its fast moving action

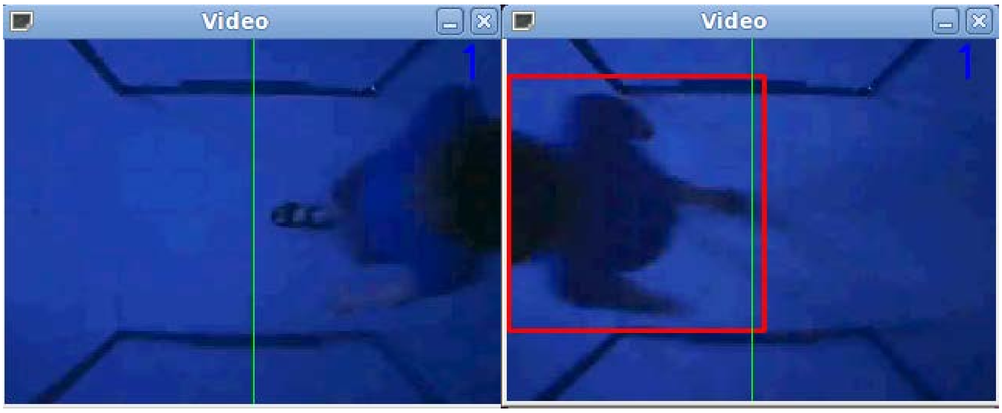
(a) (b)

(c)

Figure 5.7: Jumping video (5 FPS). In (a), fast moving object is undetected in region A and it is only detected once a large part of the moving object is in region B as shown in (b). In (c), suspicious entry warning is triggered due to the moving object's big movement when jumping though the surveillance area

(a) (b)

Figure 5.8: Side by side video from two persons situation (7 FPS). The system will detect it as only one person passed through as the two subjects have merged together from the start (a) until they passed through the surveillance area (b). Suspicious entry warning will be issued by the system due to the large contour size



(a) (b)

Figure 5.9: Low light situation (7 FPS). The moving object is only detected in one region of the surveillance area (region B) as shown in (b) causing the system to count wrongly thus successfully avoided detection

5.4 Total Computational Time for Different Frame Rate

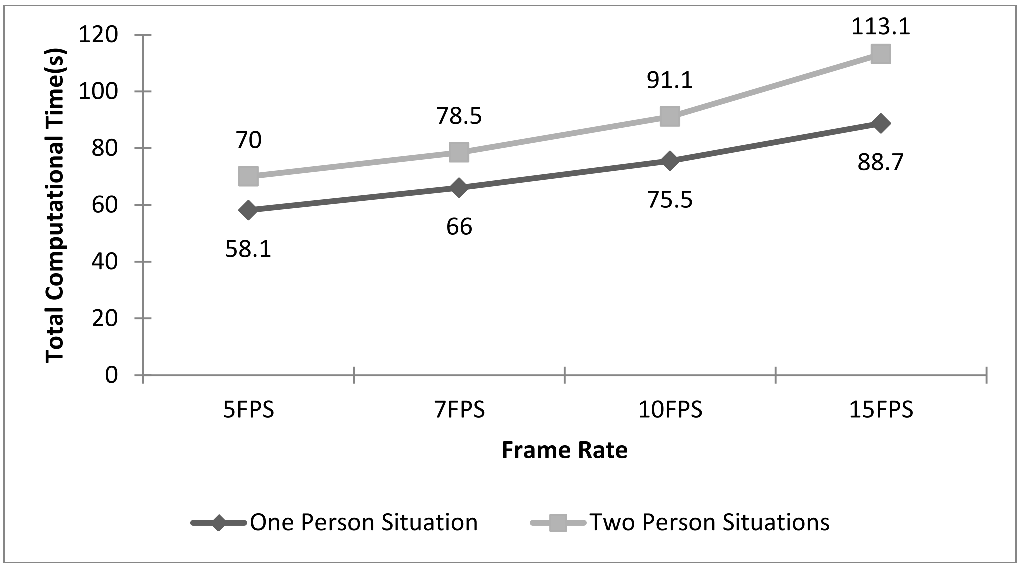


Figure 5.10: Total computational time for different FPS

To compute all time related analysis, videos consisting of one person (walking) and two persons (following closely) are used. Figure 5.10 shows the total computational time for different FPS. As expected, the total computational time increase gradually as the frame rate increase. Higher frame rate means that the system will need more time to process all the frames leading to the increase in total computational time.

5.5 Frame Rate Choosing

After computing the accuracy rate and computational time, a suitable frame rate for this system to operate on can be chosen. The following subsections show the considerations and the frame rate chosen for this system.

5.5.1 Consideration

There are a few things to consider in choosing a suitable frame rate. First, the resources in embedded system are limited compared to desktop computer based system. For example, random access memory (RAM) available on the embedded system used is limited to 512MB but it is common for a modern day desktop computer to be equipped with 2GB of RAM of higher. In this research, it is preferable to use a lower frame rate as it will result in a better performance but at the same time the accuracy rate must not be compromised. An ideal system should have a high accuracy rate with low computational time but this is not always possible especially with embedded system. Therefore, in this research priority is given to accuracy rate as a security system without an acceptable detection rate is useless no matter how fast the system is.

5.5.2 Frame Rate Chosen

Based on the accuracy and computational time test from section 5.3 and 5.4, it is found that 7 FPS is suitable for this system as it can achieve good accuracy rate for most situations while maintaining an acceptable computational time. Compared with using 5 FPS which is the lowest frame rate available, the system achieved an accuracy rate of 9.2% higher when using 7 FPS but the downside is the total computational time will increase due to higher frame rate. For security system, it is better to have a slightly slower but more accurate system as accuracy rate is still the more important aspect.

The problem with using this frame rate is that in low light situation the accuracy rate is lower compared to using higher frame rate (10 FPS and above). However, it should not be a major concern as anti tailgating/piggybacking system is usually deployed in indoor environment that has sufficient lighting.

5.6 System Performance Before and After Algorithm Optimization

This section described the performance of the system before and after the algorithm optimization explained in section 4.8 is implemented. The performance is measured by recording the computational time for each module and also the overall system.

5.6.1 Average Computational Time for Each Module Before and After Algorithm Optimization

Figure 5.11 and 5.12 shows the average time needed for the system to complete a loop of each module in one person and two persons situation. The video feed acquisition module is excluded from this result as it is only used once by the system to establish background. From both figures, it can be seen clearly that the connected component analysis module has the highest computational time in both situations. This is due to the advance morphological operation used and also the contour filtering algorithm.

The algorithm optimization has effectively improved the computational time of the connected component analysis module. In one person situation, computational time is reduced by 48.4% and in two persons situation, it is reduced by 38.5%. The improvement of computational time for the connected component analysis module will lead to the improvement of the whole system as shown in the next section.

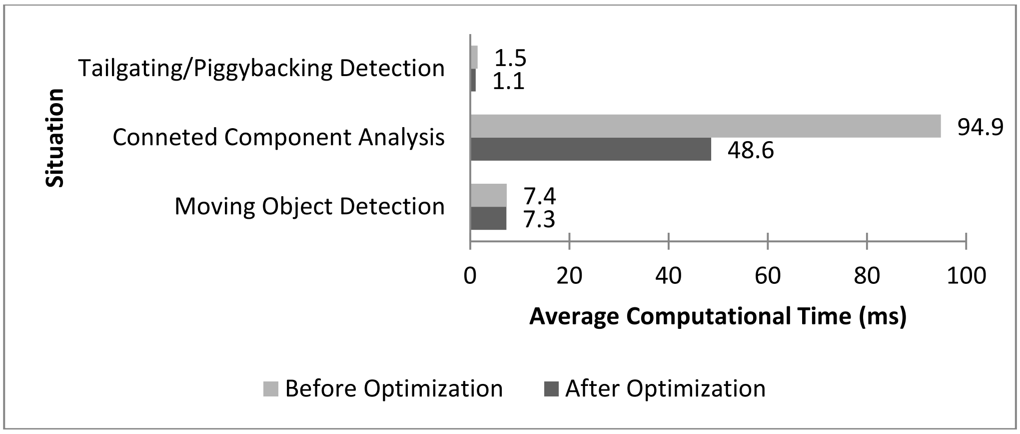


Figure 5.11: Average computational time for each module before and after optimization (one person situation)

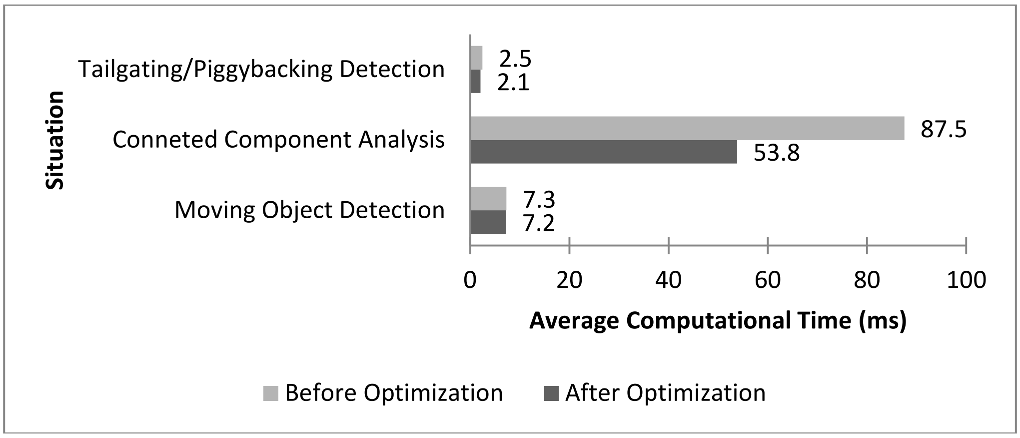


Figure 5.12: Average computational time for each module before and after optimization (two persons situation)

5.6.2 Average and Total Computational Time Before and After Algorithm Optimization

Average computational time is the average time needed for the developed system to complete a real time loop which includes moving object detection, connected component analysis and also tailgating/piggybacking detection. As seen on Figure 5.13, the average computational time for both situations has indeed improved after implementing the algorithm optimization. The computational time is reduced by 43.9% and 37.5% for one person situation and two persons situation respectively. It is important to keep the computational time of this real time loop as low as possible to ensure smooth real time analysis.

Similarly, there is also improvement on the total computational time for both situations due to the faster execution of each real time loop. The total computational time for both situations is reduced by around 18% as shown on Figure 5.14.

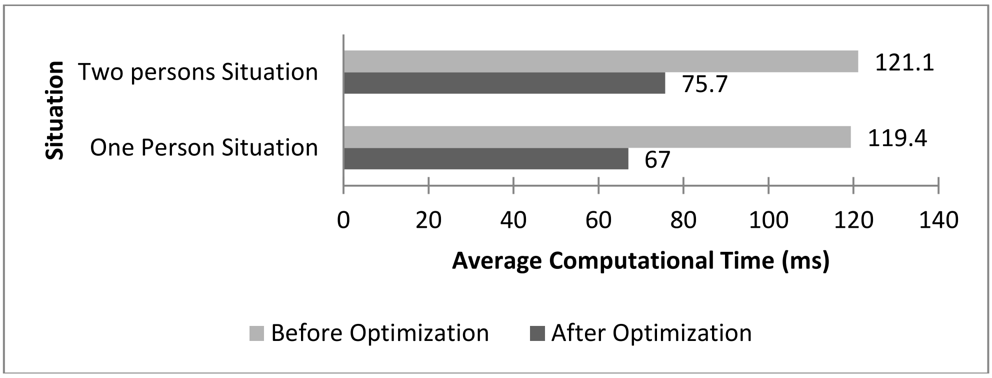


Figure 5.13: Average computational time before and after optimization

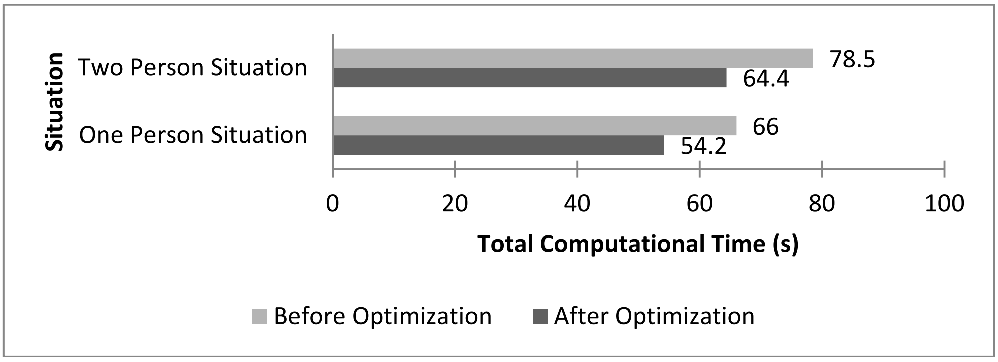


Figure 5.14: Total computational time before and after optimization

5.7 Comparison between Developed Background Subtraction Based System and Motion Template Based System

A separate system based on motion templates algorithm is developed as a comparison to the developed background subtraction based system. The following subsections show the results of this comparison.

5.7.1 System Accuracy (Motion Templates Based System)

Table 5.3: System Accuracy (Motion Templates Based System)

Situation	**Accuracy**			
	5FPS	**7FPS**	**10FPS**	**15FPS**
Walking	100%	100%	100%	100%
Running	100%	100%	100%	100%
Jumping	100%	100%	100%	100%
Sneak in	100%	100%	100%	100%
Carrying/Pushing object	100%	100%	100%	100%
Side by side	50%	50%	50%	50%
Following closely	83%	83%	83%	83%
Low light	100%	100%	100%	100%
Clothing colour similar with background	100%	100%	100%	100%
Average Accuracy Rate	**92.60%**	**92.60%**	**92.60%**	**92.60%**

From Table 5.3, motion template based system can achieve 100% accuracy rate in most situations. Unlike the developed background subtraction based algorithm, this algorithm has no problem detecting violation in running, jumping and low light situation even under the lowest frame rate. This improvement is due to different requirement when counting moving object in motion templates based algorithm. Background subtraction based algorithm requires the extraction of a complete contour of the moving object to count accurately as smaller contour will be eliminated by contour filtering algorithm. However, in challenging situations such as detecting fast moving object, the extraction of the object's complete contour is not always possible. This gives an advantage to motion templates based algorithm as it only requires part of the moving object's silhouette to start tracking.

However, motion template based system has difficulty detecting violation in two situations, one of them is the two people walking side by side situation. The system will detect it as only one people passing through under all different frame rates. The reason for false negative is because the contours of the two persons have merged together as one because they are too close together. This is the same problem faced by the developed background subtraction based system as discussed in section 5.3.

Another situation where the motion templates based system has problem detecting violation is the following closely situation. This system is not able to detect violation accurately in this situation as it does not have a contour counting stage like the background subtraction based system. Once this system computed the wrong number of people passing through, it is unable to detect violation.

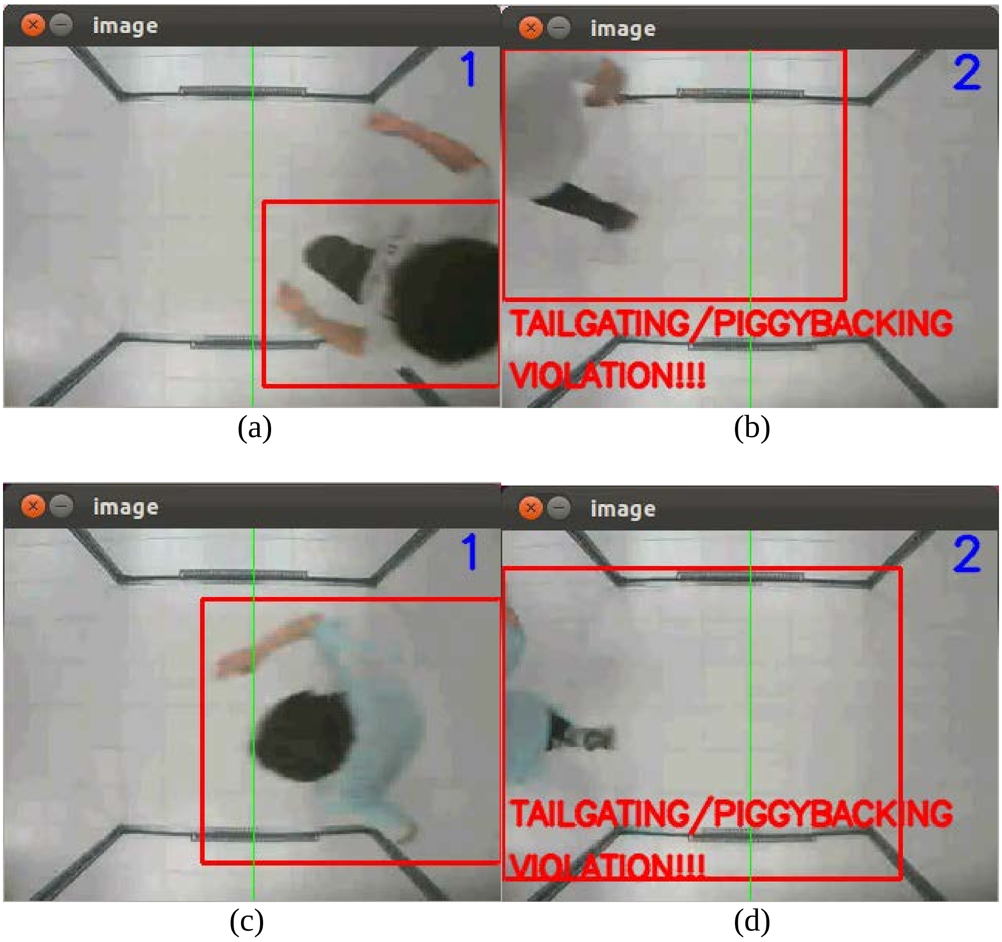


Figure 5.15: Video with fast moving objects (5 FPS). Motion template based system has no problem detecting fast moving objects even under low frame rate. (a) and (b) is running situation while (c) and (d) is jumping situation

Figure 5.16: Video in low light situation (5 FPS). Moving object under low light situation also can be detected successfully in both region A and region B by the motion templates based system even under low frame rate

(a) (b)

Figure 5.17: Video of side by side situation (5 FPS). Motion templates based system is also unable to detect violation in side by side situation as it will detect it as only one person passing through. The moving objects' contours have merged together as one big contour

(a) (b)

Figure 5.18: Video of following closely situation (5 FPS). Motion templates based system is unable to detect violation accurately in following closely situation. The system only managed to detect one person passed through in some scene when actually two people have walked past the surveillance area

5.7.2 Average Accuracy Rate

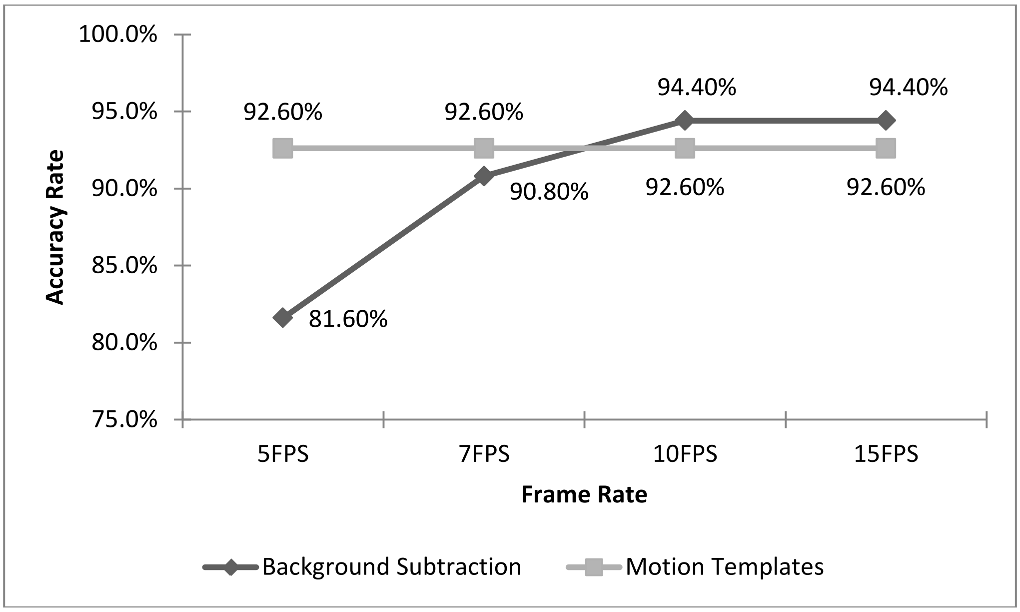


Figure 5.19: Average accuracy rate of background subtraction based algorithm and motion templates based algorithm

From Figure 5.19, it is observed that the accuracy rate for background subtraction based algorithm increase with the increase of frame rate used until it reached the maximum accuracy rate of 94.4%. Higher frame rate resulted in higher accuracy rate because moving objects will appear in more frames captured therefore lowering the possibility of moving object escaping detection especially in situation with fast moving object.

Figure 5.19 also shows that accuracy rate of motion templates based system remain constant at 92.6% under all different frame rates. This is because motion templates based system is capable of detecting violation even on less than ideal condition such as when using low frame rate (5 FPS) for video and also in low light situation. Therefore, using a higher frame rate for motion templates based system does not improves its accuracy as it has achieved its highest accuracy rate at the lowest frame rate (5 FPS).

5.7.3 Average Computational Time

In this analysis, the video frame rate used for background subtraction based algorithm is 7 FPS which is the chosen operation frame rate as describe in section 5.5.2. For motion templates based algorithm, the video frame rate used is 5 FPS as the accuracy rate for this system is the same even when using a higher frame rate.

From Figure 5.20, it is noted that background subtraction based algorithm is faster than the motion templates based algorithm even when the frame rate used is higher. Compared to motion templates based system, processing time for background subtraction based system is 4.4 times faster in one person situation and 3.8 times faster in two persons situation. The high average computational time for motion templates based system makes it unsuitable for implementation on embedded based system.

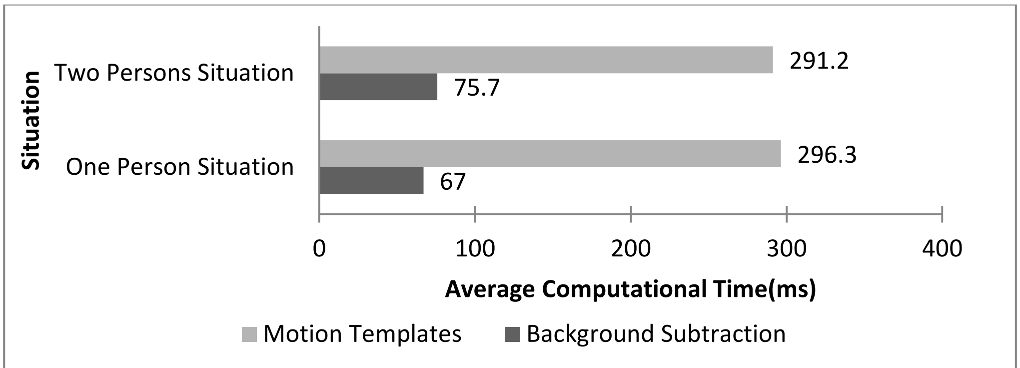


Figure 5.20: Average computational time for motion templates based system and background subtraction based system

5.7.4 Summary of Comparison between Developed Background Subtraction Based System and Motion Template Based System

From the various analyses discussed in section 5.7, it can be concluded that background subtraction based detection system is more suitable to be implement on an embedded system compared to motion templates based detection system.

1. At the chosen operation frame rate (7 FPS), accuracy rate for background subtraction based system is only 1.6% lower compared to motion templates based system but its average computational time is 75.7% faster.

2. Accuracy rate for background subtraction based system is 1.8% higher than motions templates based system at 10 FPS and above. The highest accuracy rate achieved by motion templates based system is only 92.6%.

3. Complex algorithm such as motion templates may have better performance even under low frame rate but it is computationally intensive for embedded based system. This is the main reason why a simple yet effective algorithm is preferred for embedded based system.

4. Background subtraction is one of the most basic methods in extracting moving object. However with enhancement such as real time background update and also with the addition of connected component

analysis, the quality of moving object extracted is improved and this lead to the high accuracy rate achieved by the developed system. To summarize, the performance of background subtraction based detection system is able to achieve a comparable performance similar to the more advance motion templates based detection system.

5.8 Safe Distance and Minimum Object Perimeter

Figure 5.21: Safe distance for the developed system

Safe distance is the minimum distance required between two objects to be detected as separate objects. Objects' contours will merge together as one big contour if the distance from each other is shorter than the safe distance. From various videos recorded it is found that the minimum safe distance for this system is 20-25cm. This value is found by slowly moving two objects closer to each other in the surveillance area, the minimum distance just before the two objects merged together are recorded.

The minimum object perimeter for this system is around 1.6-1.7m. The minimum perimeter is the perimeter needed for an object to be detected as a moving object by the system. This value is found by placing boxes with various sizes in the surveillance area, the perimeter of the smallest boxes that is detected by the system is measured. The minimum perimeter can be controlled by adjusting the value of preset scale in the contour filtering algorithm discussed in section 4.5.

5.9 System Limitations

Based on all the results gathered, there are two main limitations present in the system. First, the developed system is unable to detect some objects with fast moving action such as running or jumping under low frame rate (5 FPS). This problem can be solved easily by increasing the frame rate to 7 FPS but this comes at a cost as it will increase the computational time. However, the average computational time for this system can be reduced by optimizing the algorithm of the connected component analysis module. With this optimization the developed system is able to achieve a high accuracy rate while executing the detection algorithm smoothly in real time.

This system is also unable to differentiate between two persons walking side by side when passing through the surveillance area and a person carrying or pushing a large object as the system will issue suspicious entry warning for both situations. The suspicious entry warning for the side by side situation is a correct one as it is an attempt to escape detection but for carrying/pushing large object the warning is inaccurate. However, this problem has no major impact on the accuracy rate of the developed system as suspicious entry warning only served as an early warning to the security personnel and not the confirmation of a tailgating/piggybacking violation. An enhancement to improve on this situation is proposed in the next chapter.

5.10 Summary

This chapter has presented the accuracy rate of this system and analysis of the result was carried out. One of the most important findings in this chapter is that on an embedded system, the performance of the developed background subtraction based system is found to be comparable to an advance motion templates based system. It is noted that the developed system performed better than motion templates based system when using high frame rate (10 FPS and above). The advantage of motion templates algorithm is that it can perform slightly better in difficult condition such as low light situation or when using low frame rate. However, the slight increase in performance comes with a huge cost as advance algorithm such as motion templates is computationally intensive thus making it unsuitable for embedded system. Therefore, this research concludes that it is important to choose an algorithm according to the type of application and also the type of platform the developed system is going to be implemented on.

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1 Introduction

In the introduction to this dissertation, it is mentioned that tailgating/piggybacking is a serious security breach that is often not addressed in access control system. Challenges that are faced by existing anti tailgating/piggybacking systems include obstructive system that slows down crowd movement, infrared based system that can be bypassed easily and also modern machine vision based system that has a high startup cost.

Therefore, this research work aims to develop a video analytics based tailgating/piggybacking detection system that is able to detect and prevent this security issue. In the developed detection system for this research, IP camera is installed on top facing downwards to monitor violation. As a result, crowd movement will not be affected. The tailgating/piggybacking detection algorithm is developed using open source image processing library to reduce the cost of the detection system. In addition, the algorithm is implemented on an embedded system so that operating expenditure can be kept to a minimum.

Based on all the results obtained in this research, a final summary will be made in this chapter. Some possible places to deploy the developed tailgating/piggybacking detection system for this research will be discussed in section 6.4. Several ideas to further improve the system will also be put forward in the final section of this chapter.

6.2 Conclusion

From this research, a tailgating/piggybacking detection security system capable of handling different situations was successfully developed. This research utilized an inexpensive IP camera with an affordable embedded based control unit combining with various open source software.

The implementation of the detection system on embedded system has several benefits. The affordability of embedded based control unit can lower down the startup cost for the whole security system. In addition, operating cost can also be reduced due to embedded processor low power requirement. Besides that, embedded system is less prone to failure as it is designed to handle a limited number of specific functions. This will ensure continuous operation of this security system with minimal downtime.

Embedded system together with the use of IP camera as the surveillance camera provides a simpler, flexible and cost effective solution. IP camera can be easily operated by connecting it to a local server or internet and can be connected either wired for stability or wirelessly for maximum mobility.

The developed system is able achieved a high accuracy rate of 90.8% with the frame size of 320 x 240 at 7 FPS. With the three stage checking algorithm, this system is even able to outperform advance algorithm such as motion templates. The low cost and also the easy deployment of this system is certainly an attractive solution to the tailgating/piggybacking security issue compared to various existing solutions available in the market. This system can also be integrated with existing video surveillance system in a target area to provide maximum protection from different violations.

6.3 Contributions

This section described the contributions made by this research:

1. A method to optimize the connected component analysis algorithm in order to improve its performance is proposed. Connected component analysis is a computationally intensive algorithm due to its advance morphological operation and contour filtering algorithm. This will affect the performance of the developed system especially on an embedded system where resources are limited. In this research, the usage of this module is reduced by setting an ROI half the size of the surveillance area in inactive scene as explained in section 4.8. This optimization has been proven successful where computational time of the system is improved as described in section 5.6.

2. This dissertation proposed a three stage tailgating/piggybacking violation detection (section 4.7) that is able to handle various situations. The detection module utilized information that is available from the previous module (connected component analysis) such as contour position, contour size and amount of contour. By utilizing the existing information available, the detection module computational time is faster and suitable to be implemented on embedded system.

3. A comparison study between the developed background subtraction based detection system and an advance motion templates based system is done. While advance algorithm can perform better in challenging situation it might not be the suitable algorithm to be implemented on a platform with limited resources. The result of this comparison can be found in section 5.7.

6.4 Applications

Tailgating and piggybacking detection system is important to ensure only authorized personnel are allowed to enter the secured area. A few places where this kind of system will be useful are listed as follows.

6.4.1 Data Centre

A number of information technology related companies have data centre mainly used to store current or archived data. Most of these data are usually sensitive material that should only be accessible to authorize personnel. For example, privacy is a major issue in today's internet world. Any leaking of a company's customer data will violate the privacy of customers and also affect public confidence towards the company. This detection system will safeguard the security of these data by denying access to tailgater and piggybacker.

6.4.2 Residential Area

To protect the safety of residents and to create a safe environment, condominiums and gated community deploy electronic access control so that residents have to scan their access card to gain entry. Tailgating/Piggybacking violations are extremely common in these places. The authorized person will usually hold the door for the person following behind as it is basic manners. This detection system will be able to stop this practice thus preventing unauthorized personnel from gaining access to the residential area.

6.4.3 Airport/Office

Tailgating and piggybacking detection system can be installed at restricted area in both airports and offices. Security at these places especially airport is always a major concern as it involves the safety of airplanes and passengers. In addition, the installation of this detection system can also reduce operating cost as it can replace the use of guards that needed to be posted at the entrance.

6.5 Future Works

Based on the results from the previous chapter, the developed system can achieved a high accuracy rate of 90.8% but with some limitations. To overcome some of the system limitations thus further improving the system, several ideas are proposed in this section.

6.5.1 Image Processing Library Acceleration

As mentioned in chapter 3, OpenCV is the main image processing library used in this research. Therefore, most of the main functions in this detection system are built using this open source library. As OpenCV is developed by Intel, naturally its algorithm are only optimize for Intel processor. OpenCV will utilize Intel Integrated Performance Primitives (Intel IPP) to accelerate its library if the feature is found on the system (Ying 2012).

The embedded system used in this research has an ARM based processor and also a DSP (Digital Signal Processor) core. Currently all the operations are executed by the ARM processor as the OpenCV library does not utilize the DSP. One possible way to speed up the computational time is by transferring some computationally intensive OpenCV function to the DSP. Some initial work done by enthusiast has proven that, if done correctly, DSP can indeed lowered the execution time for OpenCV instruction (Poudel et al., 2010). The successful acceleration of OpenCV library will open up the possibilities of implementing more advance algorithm to further create a more robust and accurate system.

6.5.2 Head Search Algorithm

As discussed in section 5.9, one of the limitations for the developed system for this research is that it is unable to differentiate between two persons walking side by side through the surveillance area and a person carrying or pushing a large object. The system will identify both situations as suspicious entry.

To overcome this limitation, a head search algorithm can be implemented in the developed algorithm possibly using Hough transform. Hough transform is a technique to extract certain feature from an image (Shapiro and Stockman 2001) and in this case, round or oval shape which resemble a human head. Head search algorithm (Zhang and Sexton 1997; Pang and Ng 2002) can be used to differentiate these two tricky situations by calculating the number of head in the scene. In situation where two persons walking side by side, the developed algorithm will not be able to identify it as a violation because the two persons' contours have merged together as one but with the addition of head search algorithm, the system will be able to find two heads in the surveillance area and therefore issue a tailgating/piggybacking violation warning. Similarly, this same algorithm can be applied to situation where a person carrying a large object. The head search algorithm will only detect one head in this scene and the system will not identify it as suspicious entry.

In order not to overload the limited resources on the embedded platform, head search algorithm should only be used when it is necessary. One possible way to limit the use of head search algorithm is by only activating it when there's suspicious entry detected by the system. There's also no need to analyse the whole length of the video, a few frames of the video will be enough for the algorithm to calculate the number of head. Head search algorithm ultimately will be used as a verification of existing warning issued by the detection system.

In conclusion, this research demonstrates that it is possible to implement a tailgating/piggybacking algorithm on an embedded system with good accuracy.

REFERENCES

Albiol, A., Mora, I., and Naranjo, V., 2001. Real-Time High Density People Counter Using Morphological Tools. *IEEE Transactions on Intelligent Transportation Systems*, 2(4), pp. 204-218.

Axis Communications, 2012. Evolution of video surveillance systems [Online]. Available at: http://www.axis.com/products/video/about_networkvideo/evolution.htm [Accessed 30 June 2012].

Barron, J. L., Fleet, D. J., and Beauchemin, S. S., 1994. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, 12(1), pp. 43-77.

BBC, 2011. ARM Chip Promises Cheaper Smartphones. *BBC News*, [Online] 19th October. Available at: http://www.bbc.co.uk/news/technology-15372756 [Accessed 18 May 2012].

BeagleBoard.org, 2011. BeagleBoard-xM Product Details [Online]. Available at: http://beagleboard.org/hardware-xM [Accessed: 03/04/2012].

Bogomolov, Y., Dror, G., Lapchev, S., Rivlin, E., and Rudzsky, M., 2003. Classification Of Moving Targets Based On Motion And Appearance. *British Machine Vision Conference*, Norwich, 2003. pp. 429-438.

Boymond, Xavier, 2009. The surveillance control room at the Casino Théâtre Barrière de Toulouse [Online] Available at: http://www.securityinfowatch.com/news/10491598/french-casino-deploys-ip-video-solution [Accessed on 12th December 2012].

Bozzoli, M., Cinque, L., and Sangineto, E., 2007. A Statistical Method for People Counting in Crowded Environments. *14th International Conference on Image Analysis and Processing*, September 2007. pp. 506-511.

Bradski, Gary and Kaehler, Adrian, 2008. Learning OpenCV. United States of America: O'Reilly Media.

Bradski, Gary, 2012. OpenCV Wiki. [Online]. Available at: http://opencv.willowgarage.com/wiki/ [Accessed 3 June 2012].

Bramblet, J.W., Witty, C.R. and LaForge, S.M., 2008. Tailgating and Reverse Entry Detection, Alarm, Recording and Prevention Using Machine Vision. US Patent 20080285802.

Cambridge in Colour, 2012. Noise Reduction by Image Averaging. [Online]. Available at: http://www.cambridgeincolour.com/tutorials/image-averaging-noise.htm [Accessed 6 August 2012].

Canny, J., 1986. A Computational Approach to Edge Detection. *IEEE PAMI*, 8(2), pp. 679-698.

Chen, T. H., Chen, T. Y., and Chen, Z. X., 2006. An Intelligent People-Flow Counting Method For Passing Through A Gate. *IEEE Conference on Robotics, Automation and Mechatronics*, June 2006. pp. 1-6.

Davis, J. and Bradski, G., 1999. Real-Time Motion Template Gradients Using Intel CVLib. *IEEE ICCV Workshop on Frame-rate Vision*, September 1999. pp. 1-20.

Dornberger, Walter, 1954. V-2. New York: Viking Press.

Fixational, 2012. OpenCV vs Matlab. [Blog] 12 March. Available at: http://blog.fixational.com/post/19177752599/opencv-vs-matlab. [Accessed 3 July 2012].

González, R.C. and Woods, R.E., 2010. Digital Image Processing. 3rd edition. Upper Saddle River, New Jersey: Prentice Hall.

Gouaillier, V. and Fleurant, A., 2009. Intelligent Video Surveillance: Promises and Challenges [pdf]. Available at: http://www.crim.ca/fr/r-d/vision_imagerie/documents/CRIM_TDS_TechnoCommercialIntelligenceReportVideoSurveillance.pdf [Accessed 2 November 2011].

Green, W., 1999. The Appropriate and Effective Use of Security Technologies in U.S. School: A Guide for Schools And Law Enforcement Agencies [pdf]. Available at: https://www.ncjrs.gov/school/178265.pdf [Accessed 4 June 2012].

Haritaoglu, I., Harwood, D., and Davis, L. S., 2000. W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 809-830.

Heikkila, M. and Pietikainen, M., 2006. A Texture-Based Method for Modeling the Background and Detecting Moving Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), pp. 657-662.

Hou, Ya-Li and Pang, Grantham K. H., 2011. People Counting and Human Detection in a Challenging Situation. *IEEE Trans. on Systems, Man and Cybernetics*, 41(1). pp. 24-33.

Hu, Weiming, Tan, Tieniu. Wang, Liang and Maybank, S., 2004. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3), pp.334-352.

Jaimes, A. and Chang, Shih-Fu, 2000. Integrating Multiple Classifiers In Visual Object Detectors Learned From User Input. *4th Asian Conference on Computer Vision*, Taipei, January 2000.

Kettle, M., 1999. Inspectors Walk Through US Airport Security. *Guardian*, [Online] 3rd December. Available at: http://www.guardian.co.uk/world/1999/dec/03/egyptaircrash.usa [Accessed 20 July 2012].

Kim, J.W., Choi, K.S., Park, W.S., Lee, W.S. and Ko, S.J., 2002. Robust Real-Time People Tracking System for Security. *IBS Journal*, 2(3), pp184-190.

Kim, Kyungnam, Chalidabhongse, Thanarat H., Harwood, David and Davis, Larry, 2005. Real-Time Foreground–Background Segmentation Using Codebook Model. *Real-Time Imaging*, 11(3), pp172-185.

Lin, Huei-Yung and Wei, Juang-Yu, 2007. A Street Scene Surveillance System for Moving Object Detection, Tracking and Classification. *Proceedings of the IEEE Intelligent Vehicles Symposium*, Istanbul, June 2007. pp. 1077-1082.

Lipton, A.J., Fujiyoshi, Hironobu and Patil, Raju.S., 1998. Moving Target Classification and Tracking from Real-time Video. *Proceedings of IEEE Workshop on Application of Computer Vision*, Princeton, October 1998. pp. 8-14.

Lucas, B. D., & Kanade, T., 1981. An Iterative Image Registration Technique With An Application To Stereo Vision. *Proceedings of Imaging Understanding Workshop*, 1981. pp121-130.

Mackworth, N. H., 1950. Researches on the Measurement of Human Performance. Med. Res Council Spec Rep Ser, (268).

Maddalena, L. and Petrosino, A., 2008. A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. *IEEE Transactions on Image Processing*, 17(7), pp. 1168-1177.

Mathworks, 2013. System Requirements and Platform Availability by Product [Online]. Available at: http://www.mathworks.com/products/availability/ [Accessed 1 January 2013].

Mccormick, C., 2007. Setting the Mantrap. Airports International, July, pp. 32-33.

Meyer, D., Denzler, J., & Niemann, H., 1997. Model Based Extraction of Articulated Objects in Image Sequences for Gait Analysis. *International Conference on Image Processing*, October 1997. pp. 78-81.

Murali, S. and Girisha, R., 2009. Segmentation of Motion Objects from Surveillance Video Sequences using Temporal Differencing Combined with Multiple Correlation. *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, Genova, September 2009. pp. 472-477.

On-Net Surveillance Systems Inc., 2002. MJPEG vs MPEG4: Understanding the differences, advantages and disadvantages of each compression technique [Online]. Available at: http://www.onssi.com/downloads/OnSSI_WP_compression_techniques.pdf [Accessed 23 June 2012].

Pal, A., Schaefer, G. and Celebi, M.E., 2010. Robust Codebook-Based Video Background Subtraction. *IEEE International Conference on Acoustics Speech and Signal Processing*, Dallas, March 2010. pp. 1146-1149.

Pang, GKH and Ng, CK, 2002. Automated People Counting Using Template Matching And Head Search. *9th IEEE Conference on Mechatronics and Machine Vision in Practice*, Chiang Mai, September 2002. pp. 323-331.

Poudel, P., Estevez, L., Roberts-Hoffman, K. and Lira L. G., 2010. Opencv-Dsp-Acceleration. [Online]. Available at: http://code.google.com/p/opencv-dsp-acceleration/ [Accessed 5 August 2012].

Regazzoni, C. S., Cavallaro, A., Wu, Y., Konrad, J., and Hampapur, A., 2010. Video Analytics for Surveillance: Theory and Practice. IEEE Signal Processing Magazine, September 2010. pp. 16-17.

Shapiro, Linda and Stockman, George, 2001. Computer Vision. Upper Saddle River, New Jersey: Prentice Hall.

Stauffer, C. and Grimson, W.E.L., 1999 Adaptive Background Mixture Models for Real-Time Tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, June 1999.

Stonington, Joel and Gardiner, Sean, 2010. NYPD Weighs New Posts Abroad. *The Wall Street Journal*, [Online] 12th May. Available at: http://online.wsj.com/article/SB10001424052748703565804575238710598803680.html [Accessed 12 September 2012].

Tickner, A. H., and Poulton, E. C., 1973. Monitoring Up To 16 Synthetic Television Pictures Showing a Great Deal of Movement. *Ergonomics*, 16(4), pp. 381–401.

TP-LINK Technologies, 2012. Wireless 2-Way Audio Surveillance CameraTL-SC3130G. [Online]. Available at: http://www.tplink.com/en/products/details/?categoryid=230&model=TL-SC3130G#spec [Accessed 13 June 2012].

Ubuntu, 2012. Ubuntu Desktop: Features in Ubuntu 12.04.1. [Online]. Available at: https://wiki.ubuntu.com/PrecisePangolin/ReleaseNotes/UbuntuDesktop [Accessed 15 June 2012].

Velipasalar, S., Tian, Y. L., and Hampapur, A., 2006. Automatic Counting of Interacting People by Using a Single Uncalibrated Camera. *IEEE International Conference on Multimedia and Expo*, July 2006. pp. 1265-1268.

Ware, J. R., Robert, A. B., & Richard, W. S., 1964. Effect of Increasing Signal Load on Detection Performance in a Vigilance Task. *Perceptual and motor skills*, 18(1), pp. 105-106.

Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P., 1997. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), pp. 780-785.

Xu, L. Q., 2007. Issues in Video Analytics and Surveillance Systems: Research/Prototyping vs. Applications/User Requirements. *IEEE Conference on Advanced Video and Signal Based Surveillance*, September, 2007. pp. 10-14.

Ying, H., 2012. Using Intel® IPP with OpenCV 2.1. [Online]. Available at: http://software.intel.com/en-us/articles/using-intel-ipp-with-opencv/ [Accessed 8 July 2012].

Zhang, Lijing and Liang, Yingli, 2010. Motion Human Detection Based on Background Subtraction. *Second International Workshop on Education Technology and Computer Science (ETCS)*, Wuhan, March 2010. pp. 284-287.

Zhang, X. and Sexton, G., 1997. Automatic Human Head Location for Pedestrian Counting. *Sixth International Conference on Image Processing and Its Applications*, Dublin, July 1997. pp. 535-540.

APPENDIX A

Publication

1. Chan, T. W., Yap, V. V. and Soh, C. S., 2012. Embedded Based Tailgating/Piggybacking Detection Security System. *IEEE Colloquium on Humanities, Science and Engineering (CHUSER)*, Kota Kinabalu, December 2012. pp. 277-282.

APPENDIX B

Code for Tailgating/Piggybacking Detection Security System

```
#include <stdio.h>
#include "cv.h"
#include "highgui.h"

#define CVCLOSE_ITR 1
//smaller periscale means contour needs to be bigger to be
detected//
#define periScale 2
#define CVCONTOUR_APPROX_LEVEL 2

//Declaration//
IplImage *video = 0, *background=0, *currenttemp1=0, *current=0,
*frameForeground=0, *backgroundtemp2=0, *currenttemp2=0,
*backgroundtemp1=0;

int avgX=0 ;
int numppl = 0;
int numCont=0;
int regionAflag =0;
int contourflag=0;
int updatecount=0;
int areaflag=0;
double area=0;
char buffer[50];

CvMemStorage* mem_storage = NULL;
CvSeq* contours = NULL;

CvCapture* capture;
IplImage* frame = 0;

CvPoint pt1, pt2;

//function to find moving object in region A, used when ROI is
set//
void find_moving_contour(IplImage* mask, IplImage* livefeed)
{
      cvSmooth(mask, mask, CV_GAUSSIAN, 3, 3,0,0);
      cvThreshold(mask, mask, 35, 255, CV_THRESH_BINARY);
      cvMorphologyEx(mask,mask,0,0,CV_MOP_OPEN, CVCLOSE_ITR );
      cvMorphologyEx(mask,mask,0,0,CV_MOP_CLOSE, CVCLOSE_ITR );
      cvDilate(mask,mask,0,11);

      if( mem_storage ==NULL )
      {
            mem_storage = cvCreateMemStorage(0);
      }
      else
      {
            cvClearMemStorage( mem_storage );
      }
```

```
	//identify all contour//
	CvContourScanner scanner = cvStartFindContours( mask,
mem_storage, sizeof(CvContour), CV_RETR_CCOMP,
CV_CHAIN_APPROX_SIMPLE, cvPoint(0,0) );

	CvSeq* c;
	numCont = 0;
	//delete contour if contour's size is less than q//
	while( (c=cvFindNextContour(scanner))!=NULL)
	{
		double len = cvContourPerimeter(c);
		double q= (mask->height + mask->width)/5;
		if (len<q)
		{
			cvSubstituteContour(scanner,NULL);
		}
		else
		{
			CvSeq* c_new;
			c_new= cvApproxPoly(c,sizeof(CvContour),
mem_storage, CV_POLY_APPROX_DP, CVCONTOUR_APPROX_LEVEL,0);

			cvSubstituteContour(scanner,c_new);
			numCont++;
		}
	}
	contours= cvEndFindContours(&scanner);

	const CvScalar CVX_WHITE = CV_RGB(0xff,0xff,0xff);
	const CvScalar CVX_BLACK = CV_RGB(0x00,0x00,0x00);

	//draw the surviving contour back into the image//

	cvZero(mask);

	for(c=contours; c!=NULL; c=c->h_next){
	cvDrawContours(mask,c,CVX_WHITE,CVX_BLACK,-
1,CV_FILLED,8,cvPoint(0,0));
	}
}

//Connected Component Analysis//
void find_connected_components(IplImage* mask, IplImage*
livefeed)
{
	cvSmooth(mask, mask, CV_GAUSSIAN, 3, 3,0,0);
	cvThreshold(mask, mask, 35, 255, CV_THRESH_BINARY);
	cvMorphologyEx(mask,mask,0,0,CV_MOP_OPEN, CVCLOSE_ITR );
	cvMorphologyEx(mask,mask,0,0,CV_MOP_CLOSE, CVCLOSE_ITR );
	cvDilate(mask,mask,0,13);

	if( mem_storage ==NULL )
		{
		mem_storage = cvCreateMemStorage(0);
		}
	else
		{
		cvClearMemStorage( mem_storage );
		}
```

```
	//identify all contour//
	CvContourScanner scanner = cvStartFindContours( mask,
mem_storage, sizeof(CvContour), CV_RETR_CCOMP,
CV_CHAIN_APPROX_SIMPLE, cvPoint(0,0) );

	CvSeq* c;

	numCont = 0;

	//delete contour if contour's size is less than q//
	while( (c=cvFindNextContour(scanner))!=NULL)
	{
		double len = cvContourPerimeter(c);
		double q= (mask->height + mask->width)/periScale;
		if (len<q)
		{
			cvSubstituteContour(scanner,NULL);
		}
		else
		{
			CvSeq* c_new;
			c_new= cvApproxPoly(c,sizeof(CvContour),
mem_storage, CV_POLY_APPROX_DP, CVCONTOUR_APPROX_LEVEL,0);
			cvSubstituteContour(scanner,c_new);
			numCont++;
		}
	}
	contours= cvEndFindContours(&scanner);

	const CvScalar CVX_WHITE = CV_RGB(0xff,0xff,0xff);
	const CvScalar CVX_BLACK = CV_RGB(0x00,0x00,0x00);

	//draw the surviving contour back into the image//

	cvZero(mask);

	for(c=contours; c!=NULL; c=c->h_next){

	cvDrawContours(mask,c,CVX_WHITE,CVX_BLACK,-
1,CV_FILLED,8,cvPoint(0,0));

	//Bounding Rectangle on moving object//
	CvRect bndRect = cvRect(0,0,0,0);
	bndRect = cvBoundingRect(c, 0);

		pt1.x = bndRect.x;
	pt1.y = bndRect.y;
	pt2.x = bndRect.x + bndRect.width;
	pt2.y = bndRect.y + bndRect.height;

       cvRectangle(livefeed, pt1, pt2, CV_RGB(255,0,0), 2, 8, 0);

	//Check X Position and print it//
	avgX = (pt1.x+pt2.x)/2 ;

	//Calculate Contour Area//
	 area = fabs(cvContourArea(c, CV_WHOLE_SEQ,0));
		printf("%f\n", area);
				       }

}
```

```
int main( int argc, char** argv )
{


      printf("Press ESC to Close.\n");

      capture = cvCreateFileCapture(argv[1]);

      //Abort if no video//
      if( !capture )
            {
            printf("Could not initialize capturing...\n");
            }

      //create the window for the Camera Output//
      cvNamedWindow( "Video", 1 );

      //Declare Font//
      CvFont font;
      cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, 1, 1, 0, 2,
CV_AA);
      CvFont font2;
      cvInitFont(&font2, CV_FONT_HERSHEY_SIMPLEX, 0.7, 0.7, 0,
2, CV_AA);

      //Get video properties//

      frame = cvQueryFrame( capture );
      double fps = cvGetCaptureProperty (capture,
CV_CAP_PROP_FPS);

      //Create All Image//
        video = cvCreateImage( cvGetSize(frame), 8, 3 );
        background = cvCreateImage( cvGetSize(frame), 8, 1 );
        current = cvCreateImage( cvGetSize(frame), 8, 1 );
      currenttemp1 = cvCreateImage( cvGetSize(frame), 8, 3 );
        frameForeground = cvCreateImage( cvGetSize(frame), 8,
1 );
        backgroundtemp2 = cvCreateImage( cvGetSize(frame), 8,
1 );
      currenttemp2 = cvCreateImage( cvGetSize(frame), 8, 1 );
        backgroundtemp1 = cvCreateImage( cvGetSize(frame), 8,
3 );

      //convert frame to grayscale and establish background//
      cvCopy( frame, backgroundtemp1, 0 );
      cvCvtColor( backgroundtemp1, backgroundtemp2,
CV_BGR2GRAY );
      cvCopy( backgroundtemp2, background, 0 );
//currently frame in grayscale//
      cvNamedWindow( "Background", 1 );

      while(1)
      {

      frame = cvQueryFrame( capture );
        //Break if no frame//
        if( !frame )
            break;

      //make a copy for live video frame//
```

```
        cvCopy( frame, video, 0 );

        //convert current frame to grayscale//
        cvCopy( frame, currenttemp1, 0 );
        cvCvtColor( currenttemp1, currenttemp2, CV_BGR2GRAY );
        cvCopy( currenttemp2, current, 0 );


        //Background Subtraction//
        cvAbsDiff(current, background, frameForeground);

        //Set ROI to half of the surveillance area if no moving
object detected//
        if (numCont < 1){

        cvSetImageROI(frameForeground,cvRect(160,0,140,240));
                find_moving_contour(frameForeground,video);
                cvResetImageROI(frameForeground);
                }

        else {

              find_connected_components(frameForeground,video);

        //People counting//
        if(avgX>170){
                regionAflag++ ;

              }

        if(avgX<150 && regionAflag > 0)
                {
                numppl++;
                regionAflag = 0 ;
                }

        }
        //real time background update//
        updatecount++ ;
        printf("updatecount=%d\n", updatecount);
        if(updatecount>30)
              {
              printf("Checking for moving object...\n");

                if(numCont<1)
                {
                printf("Updating Background...\n");
                cvCopy( frame, backgroundtemp1, 0 );
                cvCvtColor( backgroundtemp1, backgroundtemp2,
CV_BGR2GRAY );
                cvCopy( backgroundtemp2, background, 0 );
                updatecount=0;
                }

              else {
                printf("moving object detected\n background
aborted...\n");
                updatecount=0;
                }
              }
```

```
		//Display number of ppl passed through on screen//
	sprintf(buffer, "%i", numppl);
	cvPutText(video, buffer, cvPoint(290, 25), &font,
cvScalar(255, 0, 0, 0));

	//Draw a green line on the middle
	cvLine(video,                         /* the dest image
*/
	cvPoint(160, 0),             /* start point */
		cvPoint(160, 480),            /* end point */
	cvScalar(0, 255, 0, 0),      /* the color; green */
	    1, 8, 0);

	//display violation warning on screen if people
count >1//
	if(numppl>1)
		{
		cvPutText(video, "TAILGATING/PIGGYBACKING",
cvPoint(5, 190), &font2, cvScalar(0, 0, 255, 0));

		cvPutText(video, "VIOLATION!!!", cvPoint(5, 225),
&font2, cvScalar(0, 0, 255, 0));
		areaflag=0;
		}

	//display violation warning if number of contour >1//
	if(numCont>1)
		{
		contourflag++;
		areaflag=0;
		}

	if(contourflag>1)
		{
		cvPutText(video, "TAILGATING/PIGGYBACKING",
cvPoint(5, 190), &font2, cvScalar(0, 0, 255, 0));

		cvPutText(video, "VIOLATION!!!", cvPoint(5, 225),
&font2, cvScalar(0, 0, 255, 0));
		areaflag=0;
		}

	//Display suspicious entry warning if object size is
above threshold//
	if(area>27000 && numppl<2 && numCont<2 )
		{
		 areaflag++ ;
		}
	if(areaflag>1 )
		{
		cvPutText(video, "Suspicious Entry!!!", cvPoint(10,
225), &font2, cvScalar(0, 0, 255, 0));
		}

	//displays the image in the specified window//
	cvShowImage( "Video", video );
	cvShowImage( "Background", background );

	char key = cvWaitKey(1000/fps);
	//"esc" to quit the program//
	if( key == 27 ) break;
```

```
        //"r" to reset the surveillance system//
        if( key == 114 ) {
        numppl=0 ;
        numCont=0;
        regionAflag=0;
        contourflag=0;
        areaflag=0;
        }

      }

        //Releases the CvCapture structure and destroy windows//
        cvReleaseCapture( &capture );
        cvDestroyWindow("Video");
        cvDestroyWindow("frameForeground");

        return 0;
}
```

APPENDIX C

Code for Motion Templates Based Algorithm

```
// motion templates code for video with 5fps//
//modified from OpenCV Motion Templates sample code:
motempl.c//

#include "cv.h"
#include "highgui.h"
#include <time.h>
#include <math.h>
#include <ctype.h>
#include <stdio.h>

#define CVCLOSE_ITR 1
#define CVCONTOUR_APPROX_LEVEL 2

CvSeq* contours = NULL;
CvMemStorage* mem_storage = NULL;
int numppl = 0;
char buffer[50];
int regionAflag =0;
int avgX =0;

// various tracking parameters (in seconds)//
const double MHI_DURATION = 0.01;
const double MAX_TIME_DELTA = 0.01;
const double MIN_TIME_DELTA = 0.001;

// number of cyclic frame buffer used for motion detection//
const int N = 2;

// ring image buffer//
IplImage **buf = 0;
int last = 0;

// temporary images//
IplImage *mhi = 0; // MHI
IplImage *orient = 0; // orientation
IplImage *mask = 0; // valid orientation mask
IplImage *segmask = 0; // motion segmentation map
CvMemStorage* storage = 0; // temporary storage

// parameters://
//  img - input video frame
//  dst - resultant motion picture
//  args - optional parameters

void  update_mhi( IplImage* img, IplImage* dst, int
diff_threshold )
{
    double timestamp = (double)clock()/CLOCKS_PER_SEC; // get
current time in seconds
    CvSize size = cvSize(img->width,img->height); // get
current frame size
```

```
    int i, idx1 = last, idx2;
    IplImage* silh;
    CvSeq* seq;
    CvRect comp_rect;
    double count;
    double angle;
    CvPoint center;
    double magnitude;
    CvScalar color;

    // allocate images at the beginning or reallocate them if
the frame size is changed //
    if( !mhi || mhi->width != size.width || mhi->height !=
size.height ) {
        if( buf == 0 ) {
            buf = (IplImage**)malloc(N*sizeof(buf[0]));
            memset( buf, 0, N*sizeof(buf[0]));
        }

        for( i = 0; i < N; i++ ) {
            cvReleaseImage( &buf[i] );
            buf[i] = cvCreateImage( size, IPL_DEPTH_8U, 1 );
            cvZero( buf[i] );
        }
        cvReleaseImage( &mhi );
        cvReleaseImage( &orient );
        cvReleaseImage( &segmask );
        cvReleaseImage( &mask );

        mhi = cvCreateImage( size, IPL_DEPTH_32F, 1 );
        cvZero( mhi ); // clear MHI at the beginning
        orient = cvCreateImage( size, IPL_DEPTH_32F, 1 );
        segmask = cvCreateImage( size, IPL_DEPTH_32F, 1 );
        mask = cvCreateImage( size, IPL_DEPTH_8U, 1 );
    }

    cvCvtColor( img, buf[last], CV_BGR2GRAY ); // convert frame
to grayscale//

    idx2 = (last + 1) % N; // index of (last - (N-1))th frame
    last = idx2;

    silh = buf[idx2];
    cvAbsDiff( buf[idx1], buf[idx2], silh ); // get difference
between frames

    cvThreshold( silh, silh, diff_threshold, 1,
CV_THRESH_BINARY ); // and threshold it

    cvMorphologyEx(silh,silh,0,0,CV_MOP_CLOSE, CVCLOSE_ITR );
    cvMorphologyEx(silh,silh,0,0,CV_MOP_OPEN, CVCLOSE_ITR );
    cvDilate(silh,silh,0,11);

    cvUpdateMotionHistory( silh, mhi, timestamp, MHI_DURATION );
// update MHI

    // convert MHI to blue 8u image//
    cvCvtScale( mhi, mask, 255./MHI_DURATION,
                (MHI_DURATION - timestamp)*255./MHI_DURATION );
    cvZero( dst );
    cvMerge( mask, 0, 0, 0, dst );
```

```
// calculate motion gradient orientation and valid orientation
mask//
    cvCalcMotionGradient( mhi, mask, orient, MAX_TIME_DELTA,
MIN_TIME_DELTA, 3 );

    if( !storage )
        storage = cvCreateMemStorage(0);
    else
        cvClearMemStorage(storage);

// segment motion: get sequence of motion components//
/* segmask is marked motion components map. It is not used
further*/
    seq = cvSegmentMotion( mhi, segmask, storage, timestamp,
MAX_TIME_DELTA );

// iterate through the motion components,//
// One more iteration (i == -1) corresponds to the whole image
(global motion)//
    for( i = -1; i < seq->total; i++ ) {

        if( i < 0 ) { // case of the whole image
            comp_rect = cvRect( 0, 0, size.width, size.height );

            color = CV_RGB(255,255,255);
            magnitude = 100;

        }
        else { // i-th motion component//
            comp_rect = ((CvConnectedComp*)cvGetSeqElem( seq,
i ))->rect;
            //reject very small components//
            if( comp_rect.width + comp_rect.height < 200 )
               continue;
            color = CV_RGB(255,0,0);
            magnitude = 30;

//Draw a rectangle on moving object and record the average x
coordinate//
cvRectangle(img, cvPoint(comp_rect.x, comp_rect.y),
cvPoint(comp_rect.x + comp_rect.width, comp_rect.y +
comp_rect.height), CV_RGB(255,0,0), 2,8,0);
avgX= (comp_rect.x+comp_rect.x + comp_rect.width)/2;
            }

        // select component ROI//
        cvSetImageROI( silh, comp_rect );
        cvSetImageROI( mhi, comp_rect );
        cvSetImageROI( orient, comp_rect );
        cvSetImageROI( mask, comp_rect );

        // calculate orientation//
        angle = cvCalcGlobalOrientation( orient, mask, mhi,
timestamp, MHI_DURATION);
        angle = 360.0 - angle;  // adjust for images with top-
left origin
//calculate number of points within silhouette ROI//
        count = cvNorm( silh, 0, CV_L1, 0 );
        cvResetImageROI( mhi );
        cvResetImageROI( orient );
        cvResetImageROI( mask );
```

```
        cvResetImageROI( silh );

        // check for the case of little motion//
        if( count < comp_rect.width*comp_rect.height * 0.05 )
            continue;

        // draw a clock with arrow indicating the direction//
        center = cvPoint( (comp_rect.x + comp_rect.width/2),
                          (comp_rect.y + comp_rect.height/2) );

        cvCircle( dst, center, cvRound(magnitude*1.2), color, 3,
CV_AA, 0 );
        cvLine( dst, center, cvPoint( cvRound( center.x +
magnitude*cos(angle*CV_PI/180)),
                cvRound( center.y -
magnitude*sin(angle*CV_PI/180))), color, 3, CV_AA, 0 );
      double data;
      printf("data=%f\n" , center.y -
magnitude*sin(angle*CV_PI/180));
    }
}


int main(int argc, char** argv)
{

    IplImage* motion = 0;
    CvCapture* capture = 0;

    capture = cvCreateFileCapture(argv[1]);

   if( !capture )
            {
            printf("Could not initialize capturing...\n");
            }
      IplImage* image = cvQueryFrame( capture );
      double fps = cvGetCaptureProperty (capture,
CV_CAP_PROP_FPS);

      CvFont font;
      cvInitFont(&font, CV_FONT_HERSHEY_SIMPLEX, 1, 1, 0, 2,
CV_AA);
      CvFont font2;
      cvInitFont(&font2, CV_FONT_HERSHEY_SIMPLEX, 0.7, 0.7, 0,
2, CV_AA);

  if( capture )
    {
        cvNamedWindow( "Motion", 1 );
      cvNamedWindow( "image", 1 );

        for(;;)
        {

            IplImage* image = cvQueryFrame( capture );
            if( !image )
                break;

            if( !motion )
            {
```

```
                motion = cvCreateImage( cvSize(image-
>width,image->height), 8, 3 );
                cvZero( motion );
                motion->origin = image->origin;
            }

            update_mhi( image, motion, 35 );

      //Count number of people passed through//
      if(avgX>160){
                 regionAflag++ ;

                }

      if(avgX<130 && regionAflag > 0)
                  {
                  numppl++;
                  regionAflag = 0 ;
                  }
      sprintf(buffer, "%i", numppl);

      cvPutText(image, buffer, cvPoint(290, 25), &font,
cvScalar(255, 0, 0, 0));

//Draw a virtual line on the center of surveillance area//
       cvLine(image,                         /* the dest image
*/
       cvPoint(160, 0),             /* start point */
       cvPoint(160, 480),            /* end point */
       cvScalar(0, 255, 0, 0),      /* the color; green */
          1, 8, 0);

      if(numppl>1)
            {
            cvPutText(image, "TAILGATING/PIGGYBACKING",
cvPoint(5, 190), &font2, cvScalar(0, 0, 255, 0));

            cvPutText(image, "VIOLATION!!!", cvPoint(5, 225),
&font2, cvScalar(0, 0, 255, 0));
            }

      cvShowImage( "Motion", motion );
      cvShowImage( "image", image );

      char key = cvWaitKey(1000/fps);
      if( key == 27 ) break;
      if( key == 114 ) {
      numppl=0 ;
      regionAflag=0;
      }

        }

        cvReleaseCapture( &capture );
        cvDestroyWindow( "Motion" );
      cvDestroyWindow( "image"  );
    }
    return 0;
}
```

APPENDIX D

IP Camera Specifications

Camera	
Image Sensor	1/4" Progressive scan CMOS sensor
Lens	F: 2.0, f: 4.0mm
Viewing Angle	Diagonal 67°, Horizontal 53°, Vertical 40°
Digital Zoom	10x Digital
Minimum Illumination	0.5 Lux
Video/Image	
Video Compression	Motion JPEG; MPEG-4
Frame Rate & Resolutions	Up to 30(NTSC) / 25(PAL) fps at 640x480, 320x240, 160x120
Video Streaming	Simultaneous Motion JPEG and MPEG-4 (Dual streaming)
General	
External Power Supply	5VDC, Max 3W
Dimensions(H X W X D)	3.7 x 2 .7 x 1.2 in. (96 x 58 x 31mm)