ROBBERY SCENE DETECTION:

DETECT INTRUDERS TO HOME AREA

BY

CHIN WAI KIT

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2016

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ROBBERY SCENE DETECTION: DETECT INTRUDERS TO HOME AREA____
____

Academic Session: JAN 2016

I CHIN WAI KIT___________________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ (Author's signature)

_________________________ (Supervisor's signature)

Address:

____
____
____

____
Supervisor's name

Date: _____________________

Date: _____________________

ROBBERY SCENE DETECTION:
DETECT INTRUDERS TO HOME AREA

BY

CHIN WAI KIT

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2016

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ROBBERY SCENE DETECTION: DETECT INTRUDERS TO HOME AREA**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : ___________________________

Date : ___________________________

ACKNOWLEGDEMENTS

First of all I would like to express my sincere gratitude and appreciation to my supervisor, Prof. Maylor Leung Kar Hang who has given me this opportunity to explore and research in computer vision project. He has provided me many guidance, suggestion, study material, and feedback to my works which can improve my project. He also willingly corrects my mistakes by giving many advices on how to prevent and improve my weaknesses and writing skills respectively. A million thanks to you.

Besides, I would also like to thank to my moderator, Dr. Tan Hung Khoon for his valuable feedbacks and suggestions to my project. His feedbacks have helped me to find out the problems in my system that I never think of.

Next, I want to thank to my beloved parents and brothers who always support me throughout the course. They always stay with me and assist me when I am facing any difficulties. I would not disappoint them by finishing this project using all my knowledge and skills. At the same time I will improve myself throughout this project.

Lastly, I would like to thanks all the participants who have given advices and suggestions to my project.

ABSTRACT

The security in residential area has become main issue in Malaysia due to the increasing in crime rate in Malaysia. Therefore, with the help of automated surveillance system, human can ensure their safety in their home area by monitoring their home. In this paper, an intrusion detection surveillance system is proposed to detect the home intrusion activity and alert the users and surrounding to prevent criminals committing the crimes. Recent pieces of work such as surveillance system, human action recognition, and object tracking papers are reviewed and studied. By using computer vision techniques, the automated surveillance system is achieved by analyzing the video captured using computer program. Next, the proposed system intends to track the objects throughout the video frames and determines whether an object is moving from outside area to home area. Several processes will be implemented such as area classification, background modeling, and object tracking to ensure the system performs accurately and correctly. In area classification, the video frames captured is divided into home area and outside area followed by background modeling to segment the moving object from the background. Finally, tracking techniques is applied to each of the object detected to determine whether the object from outside area has broken into the home area. In order to further improve the accuracy of the proposed system, gate analysis techniques is used to provide a hard evidence to support the intrusion case detected by the object tracking.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ARMA	Autoregressive Moving Average
CAMSHIFT	Continuously Adaptive Mean Shift
CCTV	Closed-circuit television
CDTW	Circular Dynamic-Time Warping
CHDS-DBN	Coupled Hierarchical Duration-State DBN
CHMM	Continuous Hidden Markov Model
DBN	Dynamic Bayesian Network
DTW	Dynamic Time Warping
EM	Expectation Maximization
GMM	Gaussian Mixture Model
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradient
IDE	Integrated Development Environment
JPDAF	Joint Probability Data Association Filtering
k-NN	k-Nearest Neighbor
LHS	Life-hand Side
MHT	Multiple Hypothesis Tracking
OpenCV	Open Source Computer Vision
OSAC	Overseas Security Advisory Council
PCA	Principal Component Analysis

SCGMM	Spatial-Colour Gaussian mixture model
SIFT	Scale Invariant Feature Transform
STV	Space-Time Volumes
SVM	Support Vector Machine
UML	Unified Modeling Language

CHAPTER 1 INTRODUCTION

1.1 Motivation and Problem Statement

In past few years, there have still many crime cases in Malaysia. Based on the report from Overseas Security Advisory Council (OSAC), the overall crime rate in Malaysia is high due to the dominant number of criminal cases in Kuala Lumpur. (Overseas Security Advisory Council, 2015). Moreover, the number of crime involved in Kuala Lumpur is increased as compared to last year. From the findings on the report, Malaysia is considered quite dangerous country in terms of number of criminal cases involved.

There are many different types of crimes involved in Malaysia, such as assaults, robberies, burglary, and more. Particularly, the robbery and burglary cases that occur in residential area had become more frequent. (Overseas Security Advisory Council, 2015). During the break-ins, it is difficult for the victim to call the police or alert the surrounding because most of the time, the criminals can predict what the victims can attempt. In addition, they will be monitoring victims' action to prevent them from calling for helps. In summary, residents require a surveillance system to monitor the home security in order to prevent the cases mentioned.

1.2 Surveillance System

Surveillance system was introduced to deal with the problem of break-ins robbery case. Surveillance system is a system that monitors the behavior, activities, actions, or changing environment in order to notify users about the security information. One of the most popular surveillance system is the Closed-circuit television (CCTV) or known as video surveillance that can monitor the environment captured by the camera. Every change of environment in the captured scene is monitored by the video surveillance system. At first, the video surveillance is done by human operators where large amount of real-time videos are monitored by one or more human in parallel and any crime activities happen in the video is detected by the human. Due to the limitation of the human, however, the following practice is not suitable for monitoring the home security environment. For example, it is impossible to let users monitor the video captured all the time since users have to do their own things as well. Secondly, once the criminals break into the home area when the users are not aware, it is very difficult for users to react even though the video is captured.

Therefore, in order to tackle the problems raised by the human surveillance, automatic computer video surveillance system is introduced. With the name itself, the surveillance system is done automatically by computer program to process the video frames to monitor the security environment as well as detect the criminal actions. The computer program is using the computer vision technology that analyzes the video frames captured from digital camera to obtain useful information from the video. Computer vision is a field of studies which including reconstruct, interpret, and understand the meaningful information in 3D scene from 2D image. Computer vision makes use of image processing and pattern recognition techniques to achieve its goals.

In summary, a surveillance system can greatly reduce the crimes happening in residential area and it provides convenient feature to allow users to prevent robberies. Moreover, automatic video surveillance system is the most suitable surveillance system to monitor the security of home area. Therefore, this project will develop a surveillance system to monitor the home area to detect the intruders or robbery scene automatically and alert users when intrusion happened in the users’ home area.

1.3 Project Scope

This project will develop a surveillance system that can detect intruders breaking into the home areas. The system will start to monitor when users activate the system. Besides, the system can also be built with the alarm system to alert the surrounding when the intruders break into the home area. The focus of this project is to study and understand the computer vision techniques then implement the techniques into the surveillance system.

1.4 Project Objectives

a) To develop a real-time surveillance system
   - The system aims to respond to the activity immediately by sending out alert to surrounding area when intruders break into home.

b) To provide semi-automated surveillance system
   - The system will allow users to classify the outside area and home area within the video frame in order to detect intrusion automatically.
   - The system aims to monitor and detect intrusion automatically without the assistance from human.

c) To design and implement the intrusion detection system by using following techniques to detect intrusion activity
   1. Area classification
      - The system should classify the scene captured into outside area (public space) and inside area (restricted zone) based on the user inputs.
   2. Object tracking
      - The system should able to perform multiple object tracking.
   3. Gate analysis
      - The system should obtain some important information from the gate such as the orientation of the gate in order to determine whether the gate is opened or closed.
   4. Intrusion detection

- The system should be able to make use of the information from the results of gate analysis and object tracking to determine whether the intrusion case happens.

d) To provide an user interface to allows user to use the proposed system

- The system is able to allow user to perform an initialization to let the system understand the environment captured by the camera.
- The system should display the results of intrusion detection, object tracking, and gate analysis to the user.

1.5 Impact, Significance and Contribution

The surveillance developed in this project will provide a robust intrusion detection feature in real-time manner. The system developed will helps to decrease the crime rate at home area, since it is difficult for thefts or criminals to commit the crime successfully if there is surveillance system monitoring. With this surveillance system, the security in residential area will be improved and the peoples in home area will be more comfortable and safe.

By consider some cases, when the owner of the house wish to go for vacation, but he is worrying about the house whether it will be intruded by theft or criminals. Another case is that, when the owner is inside the house and the thefts or criminals successfully intrude to the house, by the time owner realize the intrusion, the owner is too late to realize the situation because the owner might be killed or manipulated which allows the thefts or criminals to finish their robbery action. Therefore, the system can contribute to this group of people by alerting the neighbor household to help out when intrusion happens.

Beside the user perspective, the system also provides some useful techniques such as area classification and edge-based gate analysis. These two techniques allow the system to obtain extra information from the scene in order to improve the performance of the system. Moreover, addition technique such as object tracking is used to provide the intrusion information with the inputs from the results of area classification and gate analysis. Therefore, with these three techniques the home surveillance can be more reliable compared to other system that just using one technique.

1.6 Proposed Method

Home intrusion usually will lead to robbery case where the intruders tend to steal or snatch the valuable things from the owner. Therefore, the proposed method is using information from gate pattern and restricted zone intrusion to detect the intrusion. The gate information is obtained through a series of edge analysis and the restricted zone intrusion is detected by using object tracking and area classification techniques. Figure 1.1 shows the overview of the surveillance system proposed where the details of each block in the system flow diagram are discussed in Chapter 3 and Chapter 4.

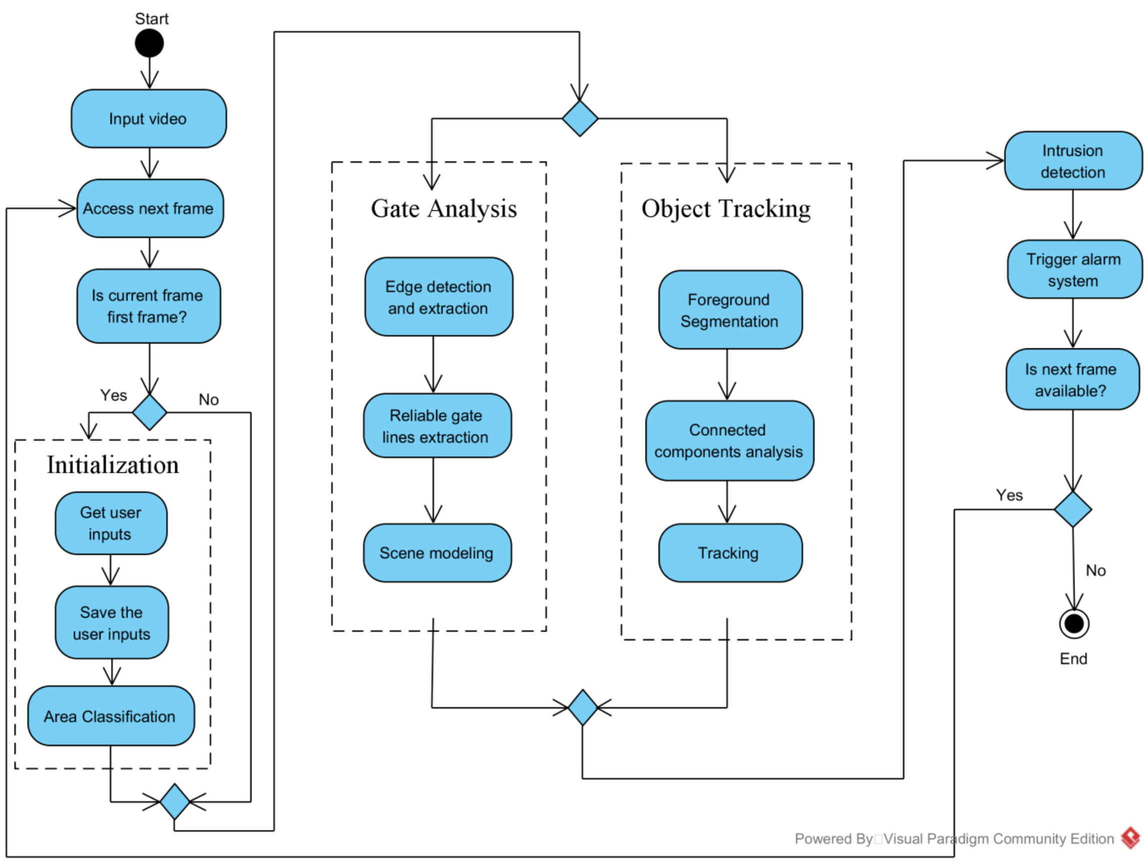


Figure 1.1 System flow diagram of proposed system

1.7 Report Organization

The rest of the paper is organized as follows. The related works are presented in Chapter 2. The proposed method is described in Chapter 3. The detail of proposed system design is explained in Chapter 4. Experimental results and discussion are conducted in Chapter 5. Lastly, Chapter 6 concludes the whole project.

CHAPTER 2 LITERATURE REVIEW

Robbery cases are committed by criminals and they usually carry out some abnormal or anomaly activity such as stealing, threatening victims, assaults to commit the crime. In order to detect the activity, human activity recognition techniques were used to classify between normal activity and anomaly activity in video surveillance system. Once the abnormal activity detected, there is a high chance that the activity is a crime.

Besides, object tracking is another important technique for surveillance system, it allows system to identify the motion of the object captured in the video. To prevent the home area entered by the intruder, object tracking was used to track whether the intruders enter to the home area from outside to inside without the permission.

This chapter includes the following materials, review of existing surveillance systems, activity recognition techniques, and object tracking techniques.

2.1 Automated Surveillance System

The automated video surveillance system has received much attention as the technology gets advanced due to its automated process that is able to replace the human operation. Video surveillance system requires the computer vision techniques that are able to segment the foreground objects from the stationary background and then perform the post-processing on the foreground object in order to further analyze the object behavior to detect the anomaly actions.

2.1.1 Vision-Based Human Tracking and Activity Recognition (Bodor et al., 2003)

Bodor et al. (2003) developed an automated surveillance system to track pedestrians and detect suspicious activities. The system first tracks each with a camera and then performs the action recognition technique to detect any danger or suspicious activity involved. The overall process for the system was described in 4 steps: human detection, human tracking, human activity recognition, and high-level activity evaluation. For human detection and tracking, mixture of Gaussians (Stauffer and Grimson, 2000) for foreground segmentation and Kalman filter were used respectively. For each pedestrian detected, a bounding box is used to surround the detected predestrian. After that, as long as the tracked pedestrian remains in the video, the pedestrian image within bounding box is captured in each frame as shown in Figure 2.1. For the next 2 steps, the system records each pedestrian position and velocity state and then develops a pedestrian's position and velocity path by using Kalman filter. On the other hand, the system will generates a bounding box (shown in Figure 2.2) for "secure area" such as expensive art display that is provided by users. The warning signal will be generated when one of the conditions in following fulfilled. The authors listed several condition such as pedestrian enters the "secure area", moves faster than normal speed, loiters around the scene, and falls down.

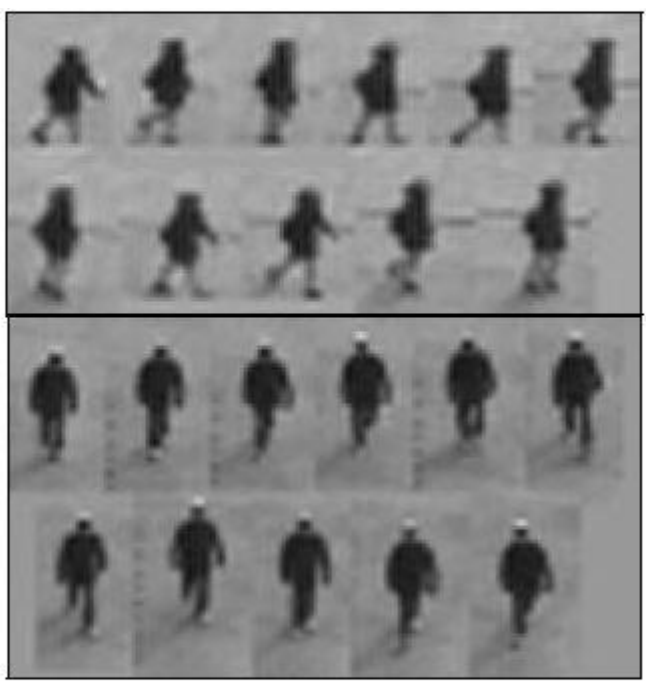


Figure 2.1 Samples image sequence of two pedestrians

Figure 2.2 Bounding box for pedestrian and “secure area”

2.1.2 Automatic Video-Based Human Motion Analyzer for Consumer Surveillance System (Lao et al., 2009)

Lao et al, (2009) introduced the combination of trajectory estimation and posture recognition to improve the human behavior analysis in their proposed system. For trajectory estimation, Lao et al. used mean shift algorithm on the persons’ appearance model such as color histogram to track the persons. For the posture recognition, the authors model the silhouettes detected by using their proposed shape descriptor called HV-PCA. Then, the authors adopt Continuous Hidden Markov Model (CHMM) for posture classification which trained by using Baum-Welch algorithm. (Rabiner, 1989). Through the classification, the classified posture can be categorized into these 5 type actions: left-pointing, right-pointing, squatting, raising hands overhead, and lying. Finally, to detect the robbery scene, an interaction model was used to understand people involved in the scene. The interaction modeling is defined as posture made by each person in the scene along the time. For example, person A has a label as “pointing”, then person B was detected as “raising hands overhead”, therefore this scene can be concluded as robbery scene.

2.1.3 W^4: Real-Time Surveillance of People and Their Activities (Haritaoglu et al., 2000)

Haritaoglu et al. present a W^4 real-time surveillance system which answers the following questions: who are the objects, what are they doing, where and when they act. The system will classify the objects detected into single person, multiple persons, and other objects for different post processing before tracking.

For single person object, the system is modeling the human posture through silhouette-based body modeling. The silhouette-based body modeling is computing the normalized projection of horizontal and vertical histogram of person silhouette as shown in Figure 2.3 and Figure 2.4. The posture is determined by the highest similarity compared to the four different training posture histograms (standing, crawling/bending, lying down, and sitting) in three different views. After that, the posture is further classified into one of the view-based appearance (front/back, left, right). If the posture of detected person is standing, the silhouette of the person undergoes the symmetry and periodicity analysis which analyze whether the standing person is carrying an object. If the person is in other postures or the person is not carrying any objects, the body parts of the person are tracked instead. The location of head is used to determine the other body parts based on the order of the body parts that defined in the posture respectively.

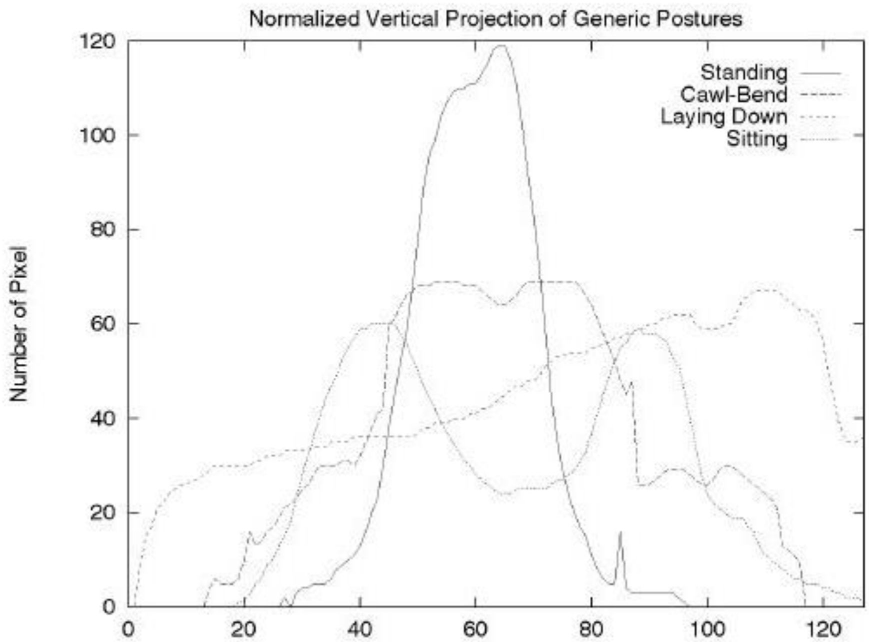


Figure 2.3 Vertical normalized projections of standing, crawling/bending, lying down, and sitting postures used in body posture estimation (Haritaoglu et al., 2000)

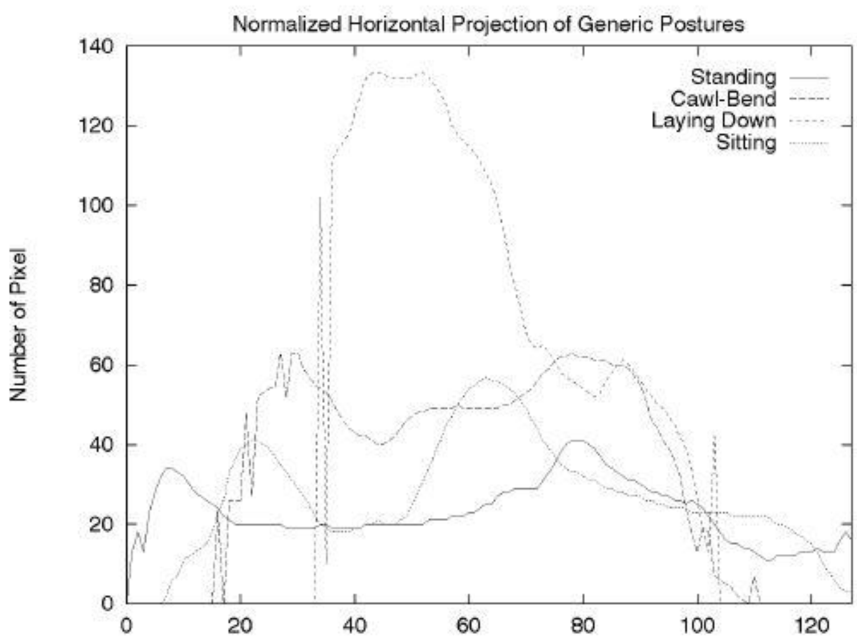


Figure 2.4 Horizontal normalized projections of standing, crawling/bending, lying down, and sitting postures used in body posture estimation (Haritaoglu et al., 2000)

For multiple persons in an object, the authors assume that every person in the group is in standing upright posture. Therefore, geometric shape cues and the vertical projection histogram of the binary silhouette are used to determine the number of head count in the group of people object. The system used intensity-based template to track each of the head detected. Using median-based estimate, the silhouette of the multiple persons object is segmented into part where each part represents a person in that group.

After that, the system will track the person detected frame-by-frame by using the median-based coordinate estimation of the position of the person and compare it with the actual position of the person to update the motion model. To further improve the system tracking performance, the system employed an appearance model which used gray-scale texture and shape to model each person for tackling the case when the person reappear to the scene after occlusion or interaction. The system continues to update the tracking information to handle the scenario when an isolated person joins to a group or when the group of person split up to a few isolated persons.

2.1.4 Discussion

Features \ Papers	Bodor et al., 2003	Lao et al., 2009	Haritaoglu et al., 2000
Camera	Static	Static	Static
Foreground Segmentation	Gaussian Mixture Model	Gaussian Mixture Model	Bimodal distribution
Number of object detected	Multiple	Multiple	Multiple
Person Recognition	-	k-Nearest Neighbor	Static shape analysis, dynamic periodicity analysis
Appearance Model	Position, velocity, and shape	Color Histogram	Shape, silhouette, projection histogram, gray-scale textural
Tracking	Kalman filter	Mean-shift, Kalman filter	Median-based coordinate estimate, binary edge correlation
Specific area warning	Yes	No	No
Action Recognition	Map of motion	Continuous Hidden Markov Model	Body part analysis

Table 2.1 Comparison of 3 reviewed surveillance systems

From table 2.1, the surveillance systems are using static camera to monitor the environment. In terms of foreground segmentation, the systems reviewed are using statistical model by generate the probability density function for each pixel to detect multiple foreground objects in each frame. Next, human object recognition can be achieved by using classification algorithm such as k-NN algorithm or shape and periodicity analysis. For appearance model, the systems use varieties of models to define the object appearance such as color histogram, shape, silhouette, and more. Color histogram might provide better model if illumination of the scene does not have significant changes while the shape of silhouette model might provide better result if the object's posture or view are not changing greatly across the frames. The selection of appearance will greatly influence on the tracking and action recognition process. With the appropriate selection of the appearance model, tracking process will be more efficient. The tracking algorithm involved in the system reviewed are Kalman filter, mean-shift, median-based coordinate estimation. Mean-shift is a template matching tracker while Kalman filter is estimation based tracker. Template matching tracker tends to have advantage on capturing object trajectory provided that the object's appearance did not change significantly. On the other hand, the Kalman filter is suitable for occlusion handling due to the pre-calculation of the object's new position. Finally, there exist many techniques involved in action recognition, but most of them perform posture analysis on the object by using different models and approaches.

2.2 Activity Recognition

Based on recent review of Ke et al. (2013), the main technology of the human activity recognition includes 3 steps as shown in Figure 2.5.

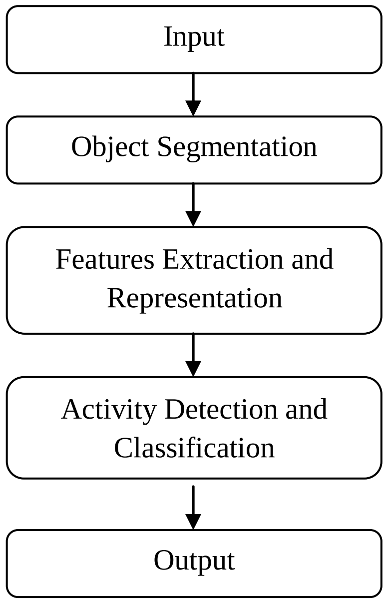


Figure 2.5 Core Technology of Human Activity Recognition

2.2.1 Object Segmentation

Object segmentation is a process that segments the foreground object from the background in order to retrieve the required objects in the frame-by-frame video. In this project, static camera segmentation is focused that the camera capture the video without changing the viewpoint of the camera such as position and angle. Therefore, a static background model is required to segment the foreground object based on the model defined. In the static camera segmentation, the following techniques are proposed in recent works which include background subtraction, statistic model, and segmentation by tracking.

2.2.1.1 Background Subtraction

Background subtraction is one of the widely used approaches for static camera segmentation due to its many different varieties of derived methods such as Running Gaussian average (Wren et al., 1997), Gaussian Mixture (Stauffer and Grimson, 1999), Eigenbackgrounds (Oliver et al., 2000) and more. Each method has its own advantages and disadvantages (Piccardi, 2004). In order to detect foreground object, the pixel's difference is calculated between the current frame and the reference frame which is the background model. When the difference calculated on a pixel of the frame above a predefined threshold, the pixel is considered to be the foreground object's pixel. However, due to its simplicity, this approach might not have good accuracy. For example, it might have difficulty to deal with the changing background. Overall, the background subtraction technique is simple yet efficient to model the background.

Running Gaussian Average

Wren et al. (1997) had proposed a technique that detects the background by considering the last n frames for every pixel by calculating the running average and standard deviation of the pixel's value. Any pixel value fall under the predefined range in the Gaussian probability density function is considered as background else the foreground. The disadvantages of this technique is that the average value is considering the past frame foreground pixel value which results in inaccuracy of the background average pixel value. In order to solve the problem, Koller et al. (1994) proposed a modified background update equation with selective background update. The boolean value is used to indicate whether the pixel is a foreground or background in order to consider the background average calculation. In short, Running Gaussian average is simple and requires less memory to model the background.

Gaussian Mixture Model

Background subtraction uses one model for a background, but Gaussian Mixture Model (GMM) uses multi-modeling to model the background in order to adapt to different scenarios. (Stauffer and Grimson, 1999). For each pixel at location (x, y), a mixture of Gaussians is used to model it. During the learning phase, the GMM is learned by using the expectation maximization (EM) algorithm. Therefore, if a pixel value is having high probability in the GMM, then the pixel most probably will be the background pixel. By defining a threshold, the image pixel will be classified as background when the value is above the threshold, or it will be classified as foreground when the value is below the defined threshold. Despite of its good performance, this approach suffered in term of computation cost from the EM. Permuter et al. (2006) proposed a new approach for GMM learning algorithm which is the k-means clustering algorithm. Although the performance from k-means clustering is not as high as the EM, the computation cost is significantly lower. Moreover, Zivkovic (2004) introduce an online adaptive GMM by using recursive equations to update the parameters. The author also state that the adaptive GMM is able to select the appropriate number of components instead of using fixed number that used by recent works. Generally, GMM can be used to describe a more complex background with high computation cost.

Eigenbackground

Eigenbackground is an eigenspace based modeling technique proposed by Oliver et al. (2000). This eigenspace model describes variety of appearances that have been observed and it provides robust model for probability distribution function of background but not moving object. This approach consists of two phases, learning phase and classification phase. In the learning phase, a sample of images is obtained to compute background image means and covariance matrix. The computed covariance matrix then undergoes the eigenvalue decomposition to obtain the eigenvector (eigenbackground). Lastly, only specific number of eigenvectors with largest eigenvalues is stored in eigenvector matrix to obtain the eigenbackground image. In the classification phase, the new image is projected into the eigenspace to remove the moving object from the background, after that it is projected back to the image space. At this point the resultant eigenbackground image contains no moving object and hence the moving object can be segmented by subtracting the original image with the eigenbackground image by fulfilling the specified threshold. Figure 2.6 shows samples of the result of eigenspace decomposition-based background subtraction technique. The limitation of this method is that it is only applicable when the background is static. Therefore, Monnet et al. (2003) and Zhong and Sclaroff (2003) addressed the limitation by introducing the methods that are able to handle the dynamic background (e.g. waving trees, water wave, moving cloud). These methods describe the image regions as autoregressive moving average (ARMA) processes that provide a way in learning and predicting the motion patterns in a scene.

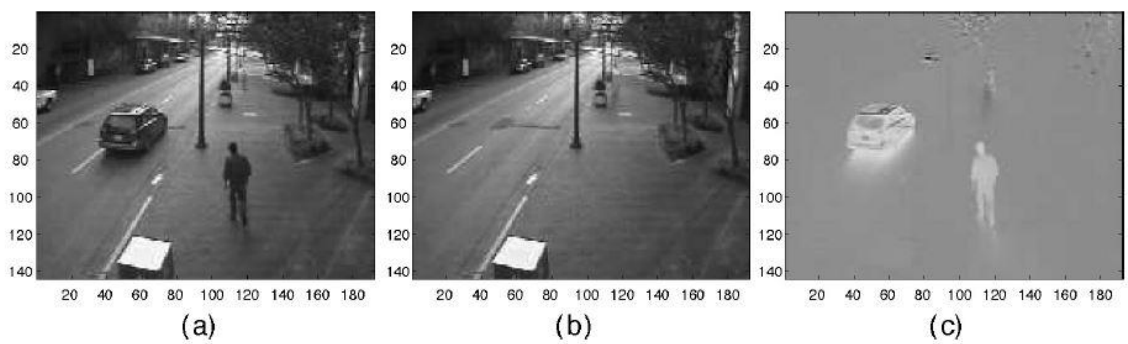


Figure 2.6 (a) Original image, (b) reconstructed image after projecting input image onto the eigenspace, (c) difference image. (Yilmaz et al., 2006).

2.2.1.2 Statistical Model

Despite the high performance from the GMM, it underperformed in brightness and chromaticity changing environment. Statistical model is another approach for static camera object segmentation that is able to model this kind of environment by using four parameters to model each pixel in the image. The four parameters are brightness distortion, chromaticity distortion, the variation of the brightness distortion, and the variation of the chromaticity distortion. (Horprasert et al., 1999). By referring the pre-defined threshold on brightness distortion and chromaticity distortion, the pixels can be classified into original background, shadow, highlighted background, or moving foreground. For example, when background distortion and chromaticity distortion are small, the pixel is classified as original background and the pixel is classified as foreground pixel if the pixel values of chromaticity distortion are high.

2.2.1.3 Segmentation by Tracking

The techniques mentioned such as background subtraction and statistical model, are deal with the pixel-level segmentation, but it will lose the global information of the object in the video frame such as the location of the object in the video frame because it only able to detect whether the pixel is consider background or foreground without consider the spatial location of the whole object. Therefore, segmentation by tracking was proposed to tackle the problem. Brendel and Todorovic (2009) proposed a segmentation by tracking regions method by using new circular dynamic-time warping (CDTW) algorithm, which matching the similar boundary portion of two regions in different frames. Furthermore, Yu et al. (2007) proposed another method which tracked the spatial-color Gaussian mixture model (SCGMM). The SCGMM is represented with five dimensions, X, Y, R, G, B where X and Y describe the spatial location and R, G, and B describe the color information of the pixel. The authors introduced an algorithm that iteratively updated the SCGMMs by fixing the spatial Gaussian models while updating the color Gaussian model and vice versa along with a constrained estimation maximization algorithm.

2.2.2 Feature Extraction and Representation

In the second phase of human activity recognition, the feature of the segmented foreground object is extracted and represented in mathematical forms for further recognition process. There are many pieces of recent work being done in this area and Figure 2.7 shows the categorization of the feature extraction and representation techniques. In global feature representation, the whole image is used to compute features, however global features are usually sensitive to noise, partial occlusion and variation of viewpoint. (Poppe, 2010). Whereas in local representation, Poppe states that local image patches and descriptors are considered as local features, and in contrast to global representation, local representation is rather invariant to changing in viewpoint, person appearance and partial occlusions.

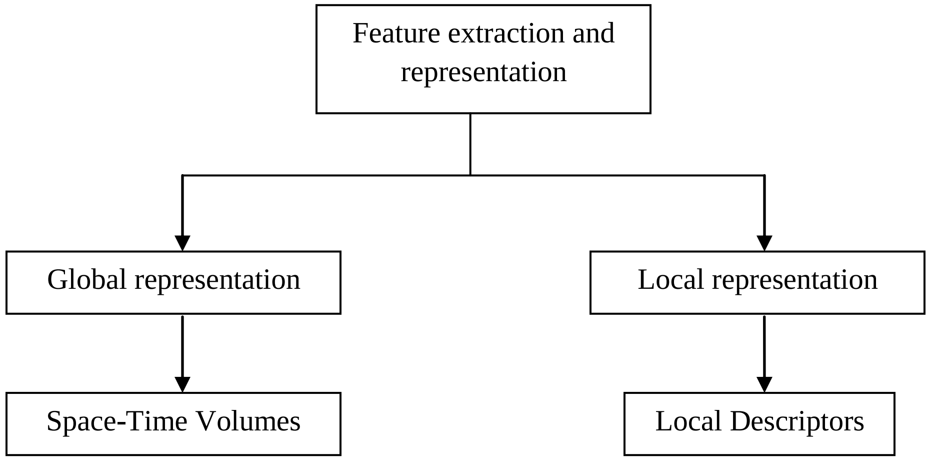


Figure 2.7 Categorization of Feature Extraction and Representation techniques

2.2.2.1 Space-Time Volumes (STV)

A 3D spatial-temporal volume is formed by stacking the video frame over a certain amount of time. The silhouettes of a segmented object are stacked over sequences of video frame to form a STV as shown in Figure 2.8. (Blank et al., 2005). The figure represents the three type of actions jumping-jack, walking, and running respectively. The x and y define the spatial information of the object and the t represents the time information. Through the SVT, local space–time saliency and orientation features can be extracted.

Figure 2.8 STV of “jumping-jack”, “walking” and “running” actions (Gorelick et al., 2007)

2.2.2.2 Local Descriptors

Local descriptors are able to handle the noises, occlusions, and possibly to rotation and scale. Scale Invariant Feature Transform (SIFT) features are local image features that are invariant to image 2D transformation, and partially to affine and illumination change. (Lowe, 1999). Lowe (2004) further explained the 4 major steps involved in generating SIFT features from image frame. Firstly, the difference of Gaussian function is applied to search over all image frames’ scales and locations to determine the interest points. For the second step, scale and location models are used to fit the scales and locations obtained, and then in the third step, more suitable points are selected as keypoints and assigned the orientation based on local image gradient directions. Lastly, the keypoint descriptor is constructed that the measured image gradients at the selected scale are transformed into a representation to allow for local shape distortion and illumination changes.

Besides SIFT, histogram of oriented gradient (HOG) descriptor is another local representation for human detection since the local object appearance and shape can usually be represented by the distribution of local intensity gradients or edge directions. (Dalal and Triggs, 2005). HOG is derived by counting the occurrences of gradient orientation in localized portions of an image. Different from SIFT, HOG is computed on a dense grid and uses overlapping local contrast normalization.

2.2.3 Activity Detection and Classification

In the third phase, activity detection and classification techniques are used to apply on the obtained feature descriptor. In the following discussion, several recent pieces of work on the activity detection and classification algorithms are explored and discussed. The following Figure 2.9 shows the categorization of the activity detection and classification techniques.

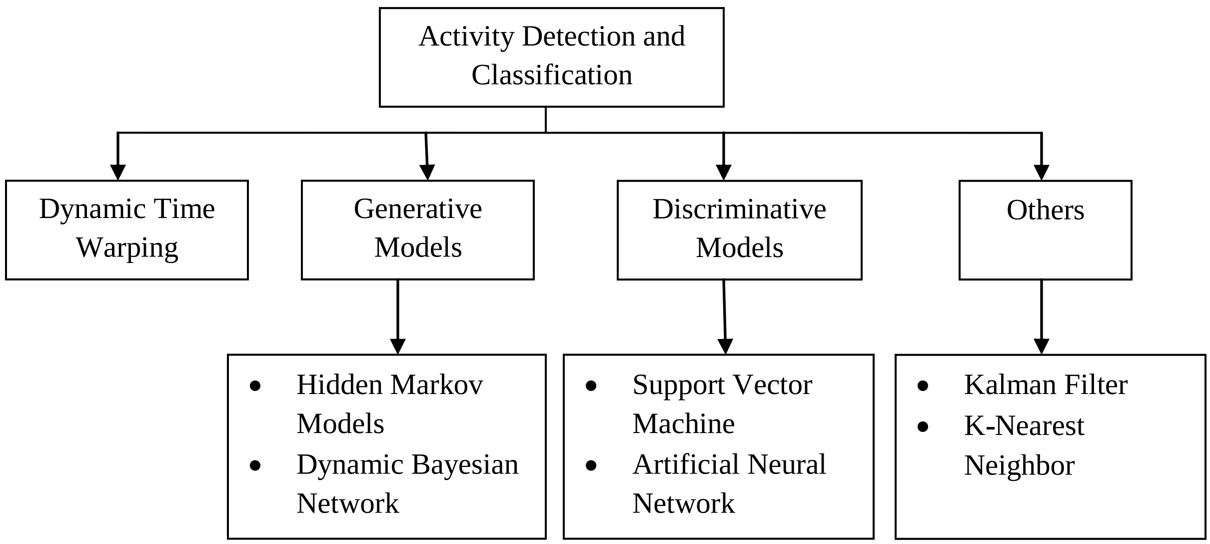


Figure 2.9 Categorization of activity detection and classification techniques

2.2.3.1 Dynamic Time Warping (DTW)

Dynamic time warping measures the similarity between 2 sequences by using the dynamic programming techniques. Veeraraghavan et al. (2005) use DTW algorithm for matching the sequences of normalized shape features. Furthermore, the DTW algorithm can be used in the extent that it can measure and recognize many kinds of human activities (Sempena et al., 2011). DTW requires many templates for matching with different scenarios which might lead to high computation cost, but it provides the speed and ease in recognize and implementation respectively.

2.2.3.2 Generative Models

Generative model is a probability and statistics model that creates the joint probability distribution by modeling each class label individually with observed variables. Given a class label, generative model will produces the likelihood of that class generates the observed variable.

Hidden Markov Models (HMM)

Hidden Markov model is defined as a doubly stochastic process which can produce the sequence of observed symbols. It consists of an underlying hidden stochastic process and an observed stochastic process. (Rabiner and Juang, 1986). It has been used popularly in temporal pattern recognition such as speech and video. (Ke et al., 2013). The hidden stochastic process and observed stochastic process are described as:

i. Hidden stochastic process (First-order Markov process)
   - Each hidden state depends only on the previous hidden state, which means that the hidden variable at time t, x_t only depends on the previous hidden state x at time $t-1$. The first-order Markov assumes that the transition probability of x_t that depends on all the past hidden states (x_1, x_2, x_3, …, x_{t-1}) is equivalent to the transition probability that depends on only the previous state x_{t-1}.

ii. Observed stochastic process
   - Each observed measurement (symbol), y depends only on the current hidden state. In other words, the observable variable at time t, y_t is only depends on the current hidden state x_t and it is independent of all other observable variables and past states.

With these two assumptions, the HMM can be illustrated where the circular nodes represent the hidden state variable and the square nodes denote the observed variables. Figure 2.10 shows the sequence of observed symbols that generated by using the hidden stochastic process and observed stochastic process.

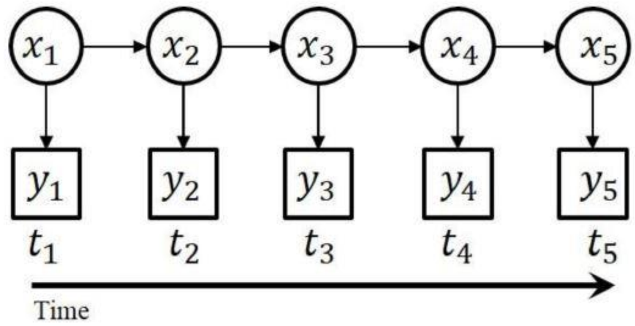


Figure 2.10 Graphical representation of HMM (Brand et al., 1997)

Dynamic Bayesian Network (DBN)

DBN is considered the generalization of HMM where it relates each random variables to others over the time. (Murphy, 2002). DBN is widely used in machine learning field to perform prediction or forecasting tasks. Besides, Coupled Hierarchical Duration-State DBN (CHDS-DBN) is proposed as an alternative for DBN. (Du et al., 2007). The CHDS-DBN is able to represent two scales of human motions, which are global activity state scale and local activity state scale. Moreover, CHDS-DBN has a low-dimensional and small size of feature space and state space respectively which is able to reduce computation complexity. It is used for modeling the interaction between human as a multiple stochastic process. However, the training for DBN is suffered from long training time.

2.2.3.3 Discriminative Models

Discriminative model determines the class labels of observed variables by learning the probabilities of class labels conditioned by the observed variables. Discriminative model tends to predict the class labels based on the observed variables which forms a decision boundary of that class.

Support Vector Machine (SVM)

The SVM proposed by Schuldt et al. (2004) is a margin-based supervised classifier in the data analysis and pattern recognition. SVM is able to construct the optimal hyperplane that can maximize the margin between two classes. The support vectors in the name are referring to the data points on the margin of the hyperplane. Schuldt et al. (2004) used the SVM to classify the human activities by extracting local space-time features in a video. Despite the good performance of SVM offer, it has a high computation cost during the learning phase.

Artificial Neural Network (ANN)

Jain et al. (2000) use a mathematical model to transform the problem into a network ANN of artificial neurons and weight directed edges. ANN is constructed in 3 layers and above, where the first layer is the input layer to receive the input from environment and the last layer is the output layer to present the processed result. All the middle layers of the ANN are hidden layers where each receives the signals of weighted neurons from the previous layers and computes the weighted sum and act as an input for next layer, the process continues until to the last layer which is the output layer. ANN proposed is used to learning the complex nonlinear input-output pairs. With sufficient training, the ANN is able to train the model to adapt to the new input. Fiaz and Ijaz (2010) proposed a method to detect human suspicious activity by using the ANN in a surveillance environment. The ANN used by the Fiaz and Ijaz is three-layer perceptron from the input of distance vectors and motion vectors for each image frame. Altough ANNs have the ability to describe complex nonlinear relationship, but it suffers from high computation cost during the learning phase and it is prone to over-fitting of the data.

2.2.3.4 Others

Kalman Filter

Kalman filter is an algorithm that using a series of observed measurements in the past along with statistical noise and other uncertainties observations to produces the estimation of Gaussian distributed state through the minimization of mean square errors. (Welch and Bishop, 1995). The algorithm is performed in two phases recursively. In the prediction phase, Kalman filter generates the estimation on the current state variables with their uncertainties. In the next phase which is update phase, the estimation from the previous phase is used to update the state of the process. Kalman filter was used to track pedestrian by generating the position and velocity path for the pedestrian. (Bodor et al., 2003). In order for Kalman filter to work, robust foreground segmentation is required to segment the object correctly.

K-Nearest Neighbor (k-NN)

The k-Nearest Neighbor algorithm measure the distance between input frames with those in a training set. The most frequent label around the k closest training data with respect to input frame is classified as the label for the input frame. Blank et al. (2005) used the global features that represent the sequence compared with the database using 1-NN (with Euclidean distance) to classify the action. Sullivan and Carlsson (2002) using k-NN to label the key poses of the tennis strokes since key poses can represent many actions instead of using sequence. One of the significant problems for k-NN classification is the k value selection. Different k value will generate different kinds of results, therefore k value should decide carefully during the training phase.

2.3 Object Tracking Techniques

The objective of object tracking is to locate the objects in every frame of the video by generating the trajectory of the object. Object tracking can be performed by either separately or jointly. For object tracking separately, the tracker corresponds object regions across every frame which is obtained by computing the means of object detection algorithm in every video frame. Besides, jointly object tracking can be achieved by iteratively estimate the next location of the object by update the object location based on the information obtained in previous frames. In order to implement a good tracker, a suitable descriptor or appearance model must be used for different kind of object. In general, object tracking technique can be categorized into 3 types which are point tracking, kernel tracking, and silhouette tracking. Each type of tracking will tackle different type of model used for the object. Figure 2.11 shows the hierarchy of the object tracking technique. (Yilmaz et al., 2006).

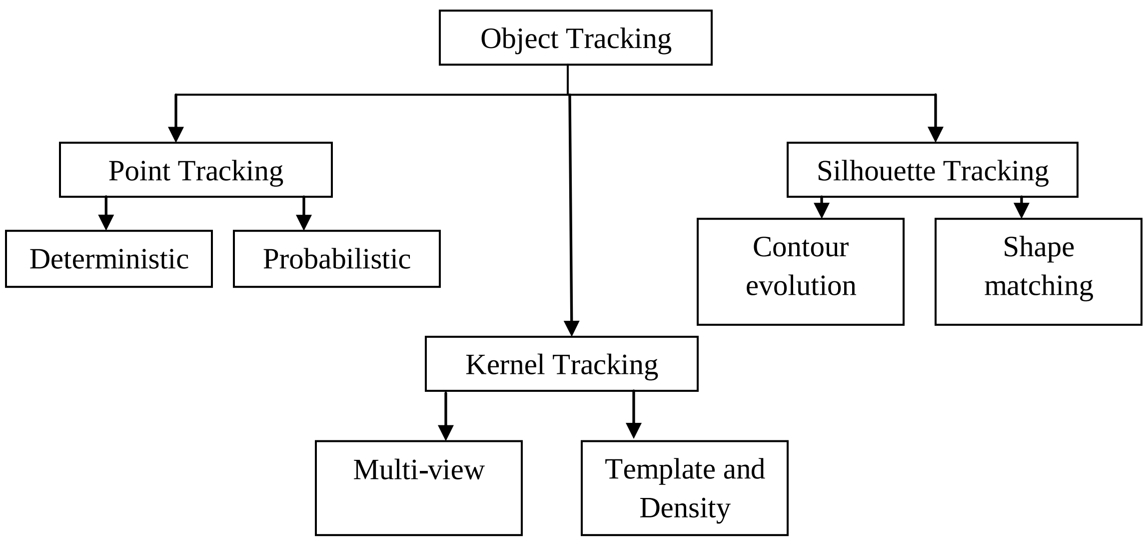


Figure 2.11 Object tracking categories

2.3.1 Point Tracking

In point tracking, the objects are represented by points and objects are tracked by building up the point correspondence using translation model.

2.3.1.1 Deterministic

Point correspondence can be formulated by defining the cost of association of each object in frame t-1 to each object in frame t where the cost of correspondence is defined by qualitative motion heuristics method. (Veenman et al., 2001). The minimization of the overall correspondence cost will produce one-to-one correspondences for each object. Sethi and Jain (1987) minimize the correspondence cost by using greedy approach on distance with neighbor points between 2 consecutive frames based on the rigidity and proximity constraints. However, this method did not handle occlusion, entry or exit of the objects. After that, Rangarajan and Shah (1991) used greedy approach as well, but with different constraints such as proximal uniformity to solve the occlusion scenario. The occlusion was handled by using constant velocity assumption to predict the position of the object. Veenman et al. (2001) improve both of the works mentioned by introduced the common motion constraint for correspondence. The proposed common motion constraint provides a strong constraint for coherent tracking of points that lie on the same object. This approach first generates the initial tracks using two-pass algorithm and the cost function is minimized by Hungarian assignment algorithm (Kuhn, 1955) in two consecutive frames. Similarly, this approach cannot handle entry or exit of object and it assume that the number of object is preserved throughout the scene.

2.3.1.2 Statistical

Different from deterministic method, statistical method consider the uncertainties into the object state estimation process instead using constraint on the object. Object in video frame are modeled using state space model which consists position, velocity, and acceleration. Broida and Chellappa (1986) used Kalman filter to estimate the optimal state variable of objects in the noisy image. This approach requires the state and noise to be distributed in Gaussian in order to perform better. Therefore, this limitation can be overcome by using particle filter. (Kitagawa, 1987). Particle filter make use of weight assigned to each sample to indicate the importance of every sample. With weight assigned to every sample, importance sampling was used to estimate the state. Both Kalman filter and particle filter only perform measurement on single object, hence joint measurement was required when tracking multiple objects.

In case for multiple object tracking, object states first required to be associated with most likely measurement of a particular object, then using the filters to estimate the next object state. Due to the varying in object position, sometimes measurement might be associated to the incorrect object which cause the filter fail to converge. Therefore, there is a need for data association technique to assign the measurement correctly. Joint Probability Data Association Filtering (JPDAF) was used to assign all measurements to each track by calculating the posterior probability of each particular measurement originated from the object associated with the particular track. (Bar-Shalom and Fortmann, 1988). The drawback for this approach is the number of track has to be fixed. Next, Reid (1979) proposed an algorithm named Multiple Hypothesis Tracking (MHT). Instead of establishing the correspondence using two frames, MHT establish the correspondence using multiple frames. Moreover, MHT was able to handle occlusion, entry, and exit of the object. MHT starts with iteration that begins with a set of hypotheses and by using each hypothesis to predict next position of each object. The predictions are compared to the actual measurements using the distance measure. New hypotheses is generated for next iteration by establish a set of correspondences for each hypothesis by referring to distance measure.

2.3.2 Kernel Tracking

The objects are modeled based on their defined shape (box or ellipse) or appearance in kernel tracking. The kernel is tracked by computing the motion of the object in the form of parametric transformation.

2.3.2.1 Multi-view based

Multi-view appearance model was used to tackle the frequently changing in the object appearance due to the different view by training the system with multi-view sample offline. Black and Jepson (1998) make use of the eigenspace to define the affine transformation of current object view to the new object view using eigenvectors. Principal Component Analysis (PCA) was used to build up the eigenspace representation of the object. The affine transformation is computed by minimizing the subspace constancy that measures the difference between the image transformed and the original image.

Besides, classification such as SVM can be applied to object tracking. (Avidan, 2004). During training, the positive examples of SVM include the image of the object to be tracked and the image of confusing background is used for negative example. The author used SVM to track the object by maximizes the classification score in image region to estimate the position of object.

2.3.2.2 Template and Density based

In single object tracking, Comaniciu et al. (2003) used a weighted histogram accumulated from the bounded object region to model the appearance of object. The authors used mean-shift algorithm to search for the most similar appearance by comparing the histograms of the object where the histogram similarity is defined by Bhattacharyya coefficient (Kailath, 1967). The mean-shift vector is computed such that the similarity is increasing and the process is done iteratively until the convergence is reached. The problem for mean-shift tracking is not able to perform well when the histograms keep changing across the time. Therefore, Bradski (1998) proposed a Continuously Adaptive Mean Shift (CAMSHIFT) algorithm which allows the tracker adapt the change in object histogram.

Besides, Jepson et al. (2003) introduced the tracker that used mixture of three components to track an object. The first component is the stable appearance component where this feature defines the reliable of slow changing pixel in the object region. For the second component which defined as transient component that describe the quick changing pixels. The last component, noise component handles the uncertainties that present in the object appearance. An online Expectation Maximization (EM) algorithm was used to train the parameters of this mixture of three components. The authors computed warping of the tracked region from current frame to next frame to get the motion of the object where the warping parameters were determine by using weighted combination of stable and transient components.

For multiple object tracking, image is modeled as a set of layers in order to tracking multiple objects. (Tao et al., 2002). Each layer is either background layer or an object layer. Each layer includes shape priors (ellipse), motion model, and layer appearance. These layers were formed by compensating the background motion with projective motion so that the object's motion can be estimated from compensated image using parametric motion. Next, probability of belonging to an object layer, p_l in every pixel is computed based on the previous object motion and shape features. If the pixel did not belong to any object, a probability of uniform background pixel p_b is assigned. The object's appearance probability p_a is then paired with p_l to obtain the final layer estimation. EM algorithm was used to estimate iteratively the 3 model parameters that maximize the observation on a layer at time *t*.

2.3.3 Silhouette Tracking

Silhouette provides a good representation for non-rigid object which cannot be achieved in simple geometric shape. The silhouette tracking is using estimation on object region in each frame by using contour evolution or shape matching.

2.3.3.1 Shape Matching

The process of shape matching is similar to the template matching in the kernel tracking in which the object silhouette and its associated model are searched in current frame. Similarity is defined as the distance between the model that generated from the hypothesized object silhouette in previous frame and current silhouette. Edge map is used as an object model for shape matching, where it is reinitialized in every frame to adapt the non-rigid object shape. (Huttenlocher et al., 1993). Hausdorff distance is used to construct a correlation surface where the minimum is selected as the new object position. (Hausdorff, 2005).

Besides edge map, silhouettes tracking can be achieved by generate silhouette trajectory from computing flow vectors for each pixel inside the object region such that the flow is dominant most of the object region. Sato and Aggarwal (2004) generate object track by applying Hough transform in the velocity space to object silhouette in consecutive frames. This approach provides motion-based matching of the object silhouettes and is insensitive to variations in object viewpoint.

2.3.3.2 Contour Evolution

Contour evolution track the silhouette by adapting the initial object contour in the previous frame to its new position in the current frame through state space model or minimization of the energy function.

In state space model, state of the objects contour is represented by shape and motion parameters. Tracking is achieved by update the object state in every instant time such that the contour’s a posteriori probability is maximized. Terzopoulos and Szeliski (1992) used dynamic of the control points (spring model) to define the object state in which the direction of control points moves based on the spring stiffness. Therefore, the new object state can be estimated through Kalman filter. In 1998, Isard and Blake used spline shape parameters and affine motion parameters to represent the object state which measure the image edges computed in the normal line to contour. The authors used particle filter to estimate the new state.

On the other hand, contour evolution can be achieved by minimizing the contour energy which defined in terms of temporal image gradient (optical flow). The optical flow constraint is depends on the brightness constancy constraint. Bertalmio et al. (2000) used optical flow constraint to evolve the current contour to new contour. Beside optical flow, Yilmaz et al., (2004) used color and texture object models to evolve the contour where the color and texture models are generated in a group around the object boundary.

CHAPTER 3 PROPOSED SYSTEM

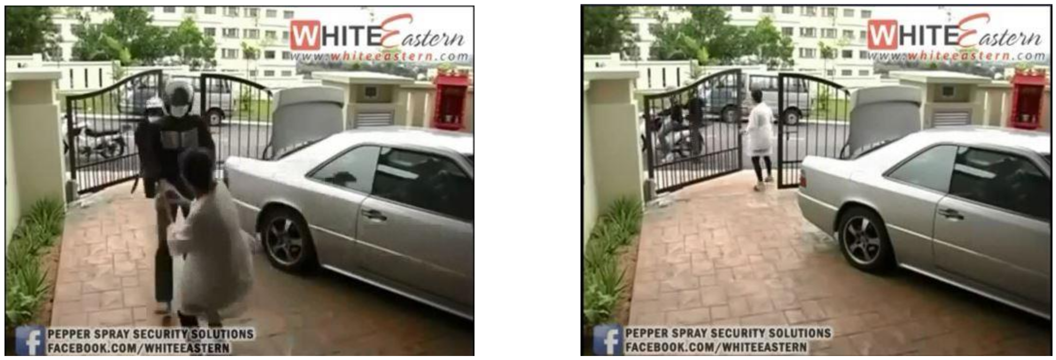
3.1 Project Scenarios

This project is to develop an intrusion detection surveillance system that able to handle the scenarios which can be illustrated by three (3) example scenarios with sequence of frames as shown in Figure 3.1, Figure 3.2, and Figure 3.3.

3.1.1 Scenario 1

a)

b)



c) d)

Figure 3.1 Sequence of frame of intrusion scenario 1

In scenario 1, the owner reached to her home and the gate was closed but it was not securely locked as shown in Figure 3.1a). In the next few frames, the intruders reached to the gate and they broke into the owner home area causing the owner panic as shown in Figure 3.1b) and 3.1c). In Figure 3.1d), the owner had lost her handbag and the intruder got ran away.

3.1.2 Scenario 2

a) b)

c) d)

Figure 3.2 Sequence of frame of intrusion scenario 2

In scenario 2, the owner was inside the house with the gate unlocked as shown in Figure 3.2a). In the next few seconds shown in Figure 3.2b), the intruders realized the unlocked gate. Then, one intruder decided to break in as shown in Figure 3.2c), luckily the owners realized the situation in time and the intruder left without success in Figure 3.2d).

3.1.3 Scenario 3

a) b)

c)

d)

Figure 3.3 Sequence of frame of intrusion scenario 3

Scenario 3 shows a woman who was cleaning the front yard in Figure 3.3a). Unfortunately, an intruder approached the owner in Figure 3.3b). In Figure 3.3c), the intruder decided to break-in to commit a crime. The owner had to surrender her valuable things to the robber in order to save her life in Figure 3.3d).

3.1.4 Discussion

Based on these three scenarios, the common approach for the intruders to break-in to the home area is by rush in through the unlocked gate due to the owner mistake. As described in scenario 2, if the owner knows the intrusion case in advance, the owner is able to put up a defence to force the robbers to retreat. Therefore, if the system is able to detect the intruders stepping into the home area and raise an alarm, the neighbor and police can come to help and make the robbers to retreat.

3.2 Design Specifications

3.2.1 Methodologies and General Work Procedures

The system proposed will segment the foreground object and detect whether there are objects from outside break into the home area. Besides, the system needs to detect the number of object appearing in the scene and then track all of them until they leave the scene. The tracked object will be labeled with a value which distinguishes the region (inside or outside the home area) of this object appear in the scene. The tracking process will continue and the label assigned to each object will be checked based on the current position of the object in the scene. When the object label is outside and it move to inside the home area, then the system will do a reasoning process to determine whether the person is an intruder. Furthermore, edge-based gate analysis will be used to provide the evidence to improve the confidence of the intrusion detection result.

3.2.2 Tools to use

i. Microsoft Visual Studio (C++)

   Integrated Development Environment (IDE) is used to develop the image processing program.

ii. Open Source Computer Vision (OpenCV)

   Library interface for Microsoft Visual Studio C++ to provide programming functions mainly aimed at real-time computer vision.

iii. Logitech HD Pro Webcam C920

   A webcam that is used to conduct experiments to evaluate the proposed system by capture sample videos

3.3 System Flow Diagram

The overall process of proposed system is described in this section. The proposed system consists of several main processes such as system initialization, object tracking, gate analysis, and intrusion detection.

Figure 3.4 shows the complete system flow which consists of the main processes mentioned in section 3.3 and some of the process is further expanded into several sub-processes to illustrate the process in detail. The assumptions of the system will be listed in section 3.4 and the detail design of the system will be presented in the next chapter.

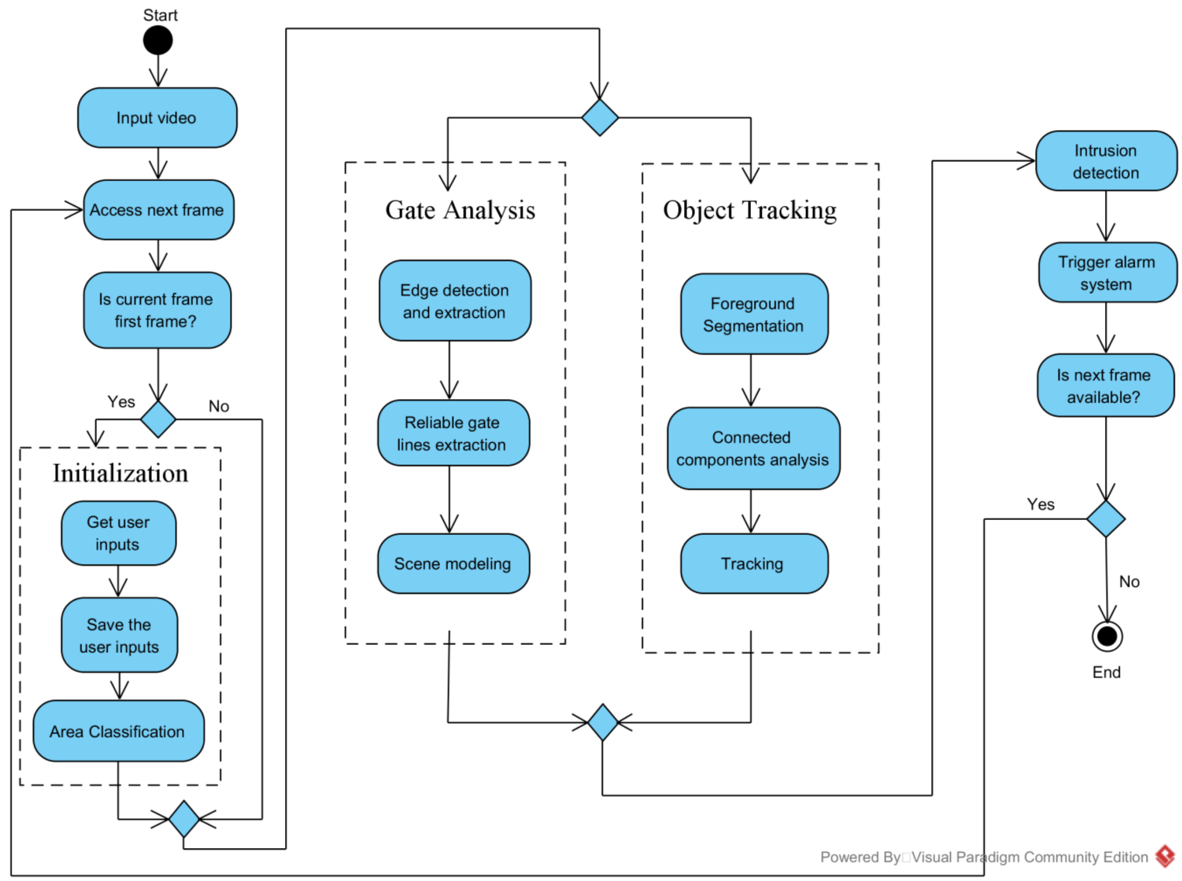


Figure 3.4 System flow diagram

3.4 Assumptions

There are several assumptions required to be considered to make sure the system work properly:

i. The system will start monitoring when users start the system manually.
ii. The system will stop monitoring when users stop the system manually.
iii. The camera needs to capture the entire main gate region in home area.
iv. The main gate must be opaque.
v. The camera needs to be placed in higher position to capture inside and outside the home areas.
vi. Intruders only break in by moving from outside to inside the home area through unlocked gate.
vii. The position of the main gate captured is required to be almost upright.

3.5 Implementation Issues/Challenges

The image complexity can be as high as possible due to the large variation in image content. Furthermore, every image or video that is captured in different places and at different time is totally different. Moreover, a 3D real life environment only can be described in 2D form in the image in which the image processing require to retrieve the information in 3D based on the 2D image. Therefore, computer vision tasks are always challenging for researcher.

CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Initialization

Before the proposed system operates, several initializations are required to obtain information from users about the scene captured. This initialization process needs user to select four (4) gate points that respect to the four (4) gate corners in the scene as shown in Figure 4.1.

Figure 4.1 Four (4) gate corners in the scene captured

After 4 gate points has been selected, the next information required is the gate region. Gate region is defined by a rectangle that can cover the whole gate region in a scene. Users are required to draw the rectangle that can cover the gate region as shown in Figure 4.2.

Figure 4.2 Rectangle that covers the gate region

Once the user finish input the information required, the system will store the user inputs for further processes which will be discuss in section 4.2. The system initialization process allows the proposed system adapt to different environment based on user inputs. However, the inputs provided by the user will acts as a basis on how the system performs. Therefore, the performance of the system will be greatly affected by the users' input.

4.1.1 Area Classification

Area classification is a process that the scene captured is divided into two (2) areas, "inside" and "outside" based on gate points selected by the user. During system initialization, gate point 1 and gate point 2 will be form a line. The line indicates the boundary between inside and outside area as well as providing the orientation of the closed gate which will be used for gate analysis in later stage.

When users select the first two gate points (point 1 and point 2), the coordinate for these two points (x_1, y_1) and (x_2, y_2) is stored in the system permanently or until users wish to change it. The line is form by using straight line equation (in slope-intercept form) as shown:

$$(y - y_1) = m * (x - x_1) \text{ , where } m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

Consider Figure 4.3, where the image is in 2D matrix form and each block is a pixel then two gate points are selected which are represented by blue circles. These coordinates for these two gate points are (9, 3) and (0, 4). Therefore, the straight line equation is form as follows:

$$m = \frac{(3-4)}{(9-0)}$$
$$= -\frac{1}{9}$$

$$(y-4) = m * (x-0)$$

$$=> y = -\frac{x}{9} + 4 \qquad (1)$$

$$=> y + \frac{x}{9} - 4 = 0 \qquad (2)$$

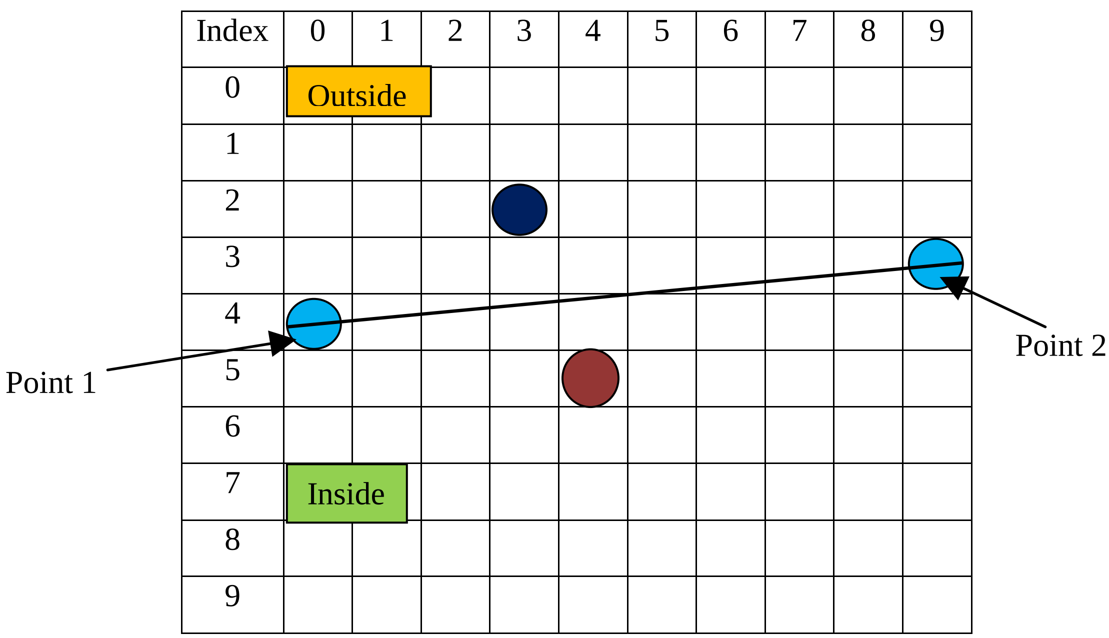


Figure 4.3 Image in 2D matrix form with area classification

Given a point (3, 2) as shown in dark blue circle in Figure 4.3, the y coordinate will be 2 and the x coordinate will be 3. By substitute these values in the equation (2), the left-hand side (LHS) of the resultant equation is less than zero (< 0), then the point is in outside area. On the other hand, if the point is below the line as shown in brown circle in Figure 3.5, the left-hand side (LHS) of the resultant equation will be more than zero (> 0). Therefore, based on the equation (2), all the pixels will be classified into outside area if its' coordinate result in LHS of the equation is less than zero (0). In contrast, if coordinate of the pixels result in LHS of the equation more than zero (0), then the pixels consider inside. The labels of "inside" and "outside" are shown in Figure 4.3.

4.1.2 Image Frame Resize

In order to allow the system work correctly, the video frame size need to be adjusted to a suitable range such that the accuracy and speed of the system can be improved. During the system initialization, the ratio for the frame scaling is required. To obtain the ratio, the following steps are used:

1. Initialize variable *i* with 1
2. Create a temporary copy of frame
3. The temporary frame width and height are divided by *i*
4. If the resultant width or height of the temporary frame is lower than 200 pixels, the loop stopped
5. Else increase *i* by 1 and go to step 1

After the loop, the variable *i* will be stored for the next frame and act as a divisor to scale down the frame image.

4.2 Gate Analysis

Gate analysis is a process that extracts the edge information from the region defined by the users and hence determines the status of the gate (open or close). The gate status can be obtain by analyze the edge information from the gate region defined by user.

4.2.1 Edge Detection and Extraction

In order to determine whether the gate is opened or closed, the orientation of the gate and the line formed by area classification are required. When the difference in gradient between the line formed by area classification and gate line is high, then the gate is consider as open and close when the difference is low as shown in Figure 4.4. In Figure 4.4, the light blue line represents the area classification line and the dark blue and yellow colors represent the close gate line and open gate line respectively.

Figure 4.4 Gate line during opening and closing

Based on the case mentioned, the edge information is useful to obtain the gate lines. Therefore, the edge information is extracted from the gate region by using the reliable edge detection algorithm known as Canny Edge Detection. (Canny, 1986). When Canny edge detection is completed, the resultants binary image with all the edge points as white pixels and the non-edge points are produced as black pixel as shown in Figure 4.5b). After that, the boundary following technique is used to locate all the edge points in the Canny Edge Detection result to generate list of contours. For each of the contour detected, polygon approximation technique is used to reduce the contour information while preserve most of the important information in the contour. By doing so, the speed of processing these contours is increase since the information is lesser. Figure 4.5c) and Figure 4.5d) show the result of boundary following (or find contour) and polygon approximation of the canny edge result. In boundary following results, each contours represented by a unique color, which shows that all the edge points has been retrieved as a list of contours.

a) Original frame with gate region

b) Canny edge results on gate region

c) Boundary following results

d) Polygon approximation

Figure 4.5 Image frame with original frame, canny edge detection result, boundary following result, and polygon approximation result respectively

4.2.2 Reliable Gate Line Extraction

After edge features are extracted from the gate region, several processes are required to obtain the gate lines that can used to infer the status of the gate. Firstly, the reliable gate lines need to be obtained followed by collecting all the gate lines that can determine the orientation of the gate. In order to determine whether the lines are reliable gate lines, several characteristics of the lines must be satisfied as follows:

- The line will remain its orientation throughout the process of opening and closing
- The line will remain its position throughout the process of opening and closing
- All type of gates must have this reliable line

With these characteristics, the reliable gate line is defined as the line that refers to the pickets that closest to the gate post as shown in Figure 4.6. Figure 4.6 shows the lines are appear to be almost the same throughout the process of opening and closing of gate.

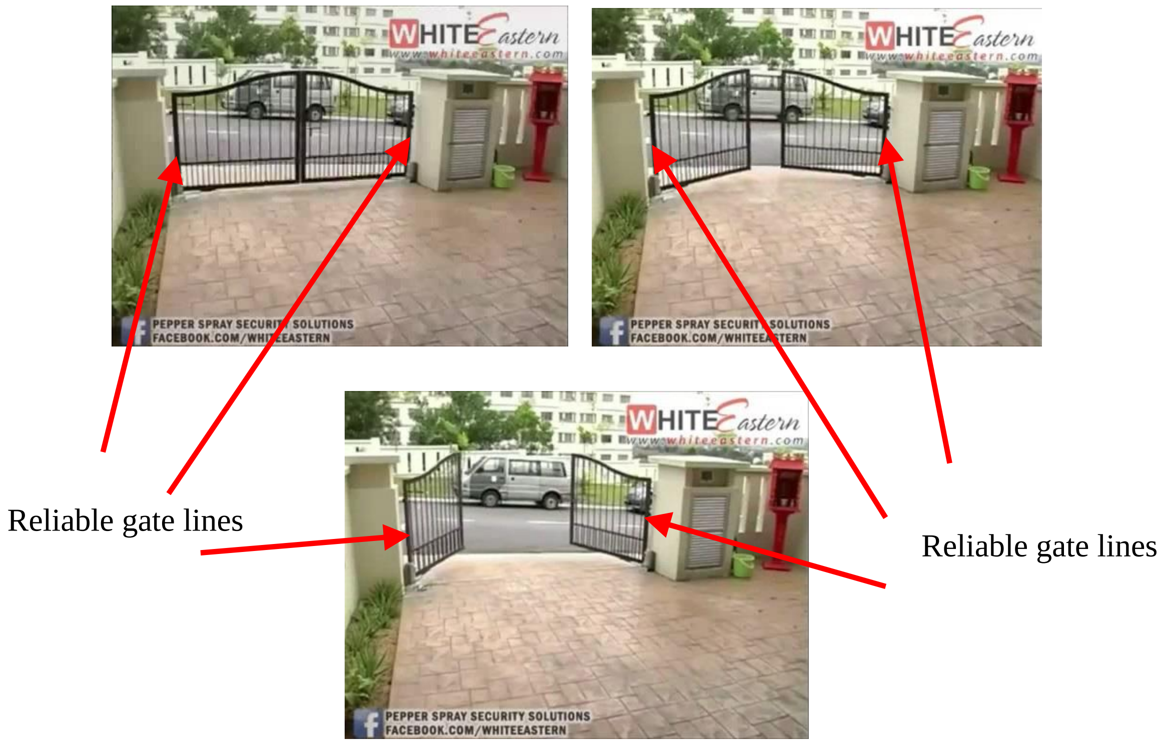


Figure 4.6 Image frame with reliable gate lines

After the defining the reliable lines, these lines are extracted out from the result of polygon approximation. In order to extract these lines, several conditions need to be satisfied. First, these reliable gate lines all have similar gradient which is close to the gradient of vertical line where the gradient is computed as shown in equation (3) below.

$$gradient = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

Therefore, in step 1, for each of the line produced from polygon approximation, compute its gradient, if the absolute of gradient is more than predefined threshold (or gate gradient threshold), then the line is consider a candidates lines for step 2. The predefined threshold used for candidates lines classification is 3.0 because the gate captured might not be always in exactly upright position. In the next step, check if the end points of candidates lines close to one of the 4 gate points. If the end points of line are close to point 1 or point

3 (in Figure 4.1), then it is in left-side of the gate. Similarly, if the end points are close to point 2 or point 4 (in Figure 4.1), then it is in right-side of the gate. All other candidate lines from step 1 that not fulfill one of the two conditions mentioned in step 2 are eliminated. After filter out the candidate lines, the checking is to make sure the candidate line is within the range of the gate points in term of y-axis. For example, the y value for both end points in each candidate lines detected must be greater than y value of point 3 (for left-sided) or point 4 (for right-sided) and at the same time it must be less than the y value of point 1 (for left-sided) or point 2 (for right sided). In order to prevent the elimination of the candidate lines that having y value is slightly less the minimum boundary value or slightly more than the maximum boundary value, an extra value is added to range such that the reliable gate can be obtained correctly. Therefore, the reliable lines are obtained as shown in yellow lines as shown Figure 4.7.

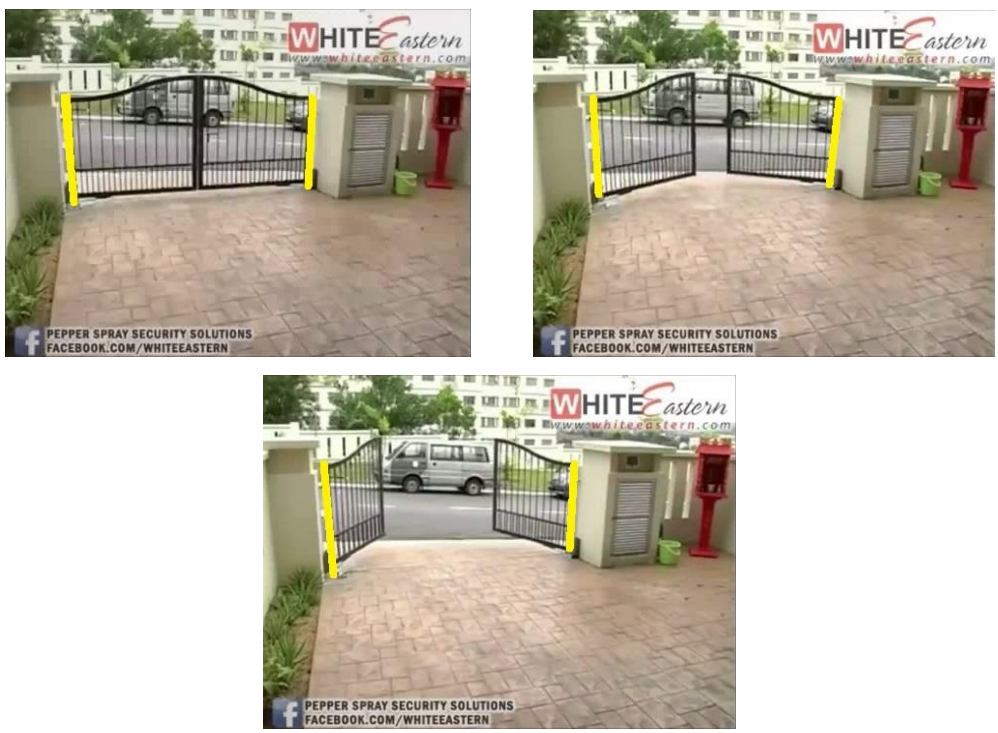


Figure 4.7 Results of reliable gate lines extraction

The reliable gate line detection process will able to work perfectly in most of the scenario except that it only fails in 3 scenarios as follows:

- The vision of reliable gate line is completely blocked by an object

- The vision of reliable gate line is blocked by an object which have the similar edge information
- The edge information is from Canny edge detection is very less

Despite the 3 scenarios that cause the process fail, the process can be used even when the reliable gate line is partially blocked. Figure 4.8 shows the reliable gate line detected using the method proposed when the gate is partially blocked.

Figure 4.8 Reliable gate lines extraction when partially blocked

4.2.3 Scene Modeling

The last part of the gate analysis is scene modeling, scene modeling is a process to model the scene that captured by surveillance camera through analysis of the sequence of video frames in order to produce useful information to the user. In this project, scene modeling mainly focuses on the gate status where the results from reliable gate lines extraction will be used to infer the status of the gate.

In Figure 4.8, the gate points (from Figure 4.1) that close to the any two of the reliable lines' end points will be used for gate lines detection whereas the points which do not have any corresponding line match are labeled as missing gate points. By using the accepted gate points, the gate lines will be detected by fulfill 2 criteria as follows:

- The gradient of the gate lines detected must be lower than the gate gradient threshold

- The gate lines detected must be close to the accepted gate points

Figure 4.9 shows the gate lines detected by using reliable gate lines and gate points, where the yellow lines, orange cross, red cross, and dark brown lines represent the reliable gate lines, accepted gate points, missing gate points, and detected gate lines respectively.

Figure 4.9 Gate analysis results

Next, for each of the detected gate lines, its gradient is computed by using equation (3) in section 4.2.2 and compare with the gradient of area classification line. If the absolute difference between absolute gradient of a gate line and absolute gradient of area classification line is greater than the minimum threshold, then the gate line is labeled as opened, else label it as closed. The minimum threshold value mentioned is varies with different type of gate. After observing the results by making several adjustments on the threshold value, the value range from 0.1 to 0.2 are having the better results. In some cases, due to the consistent performance of canny edge detection algorithm, sometimes the gate line can be detected as 2 or more lines with similar orientation or detected as shorter line due to the thickness of the pickets or rails of the gate and the type of the gate as shown in Figure 4.10.

Figure 4.10 Multiple lines detected for each gate lines

With the aforementioned problem, the status of the gate can be inferred by using counting technique. For each of the detected gate lines, calculate the number of line with open label and close label separately for each side of gate. Then, pick the highest count to update the gate status. In case of tie in the count between number of open label and close label, the gate status remains unchanged. For example, Figure 4.11 shows that there are 3 opened gate lines and 1 closed line detected for right-side gate. By using the counting techniques, since 3 is greater than 1, therefore the status of right-side gate is detected as opened.

Figure 4.11 Results of gate analysis with opened and closed gate lines

4.3 Object Tracking

The object tracking techniques is able to locate moving objects over time that appears in the scene captured by camera. The process mainly involves 3 major steps which are foreground segmentation, connected components analysis, and tracking.

4.3.1 Foreground Segmentation

Foreground segmentation is a process that segments the foreground objects from the background in a scene. Usually, a robust background model is required to segments moving objects correctly. By computing the difference between the pixels in current frame and the background model, the foreground object pixels can be segmented from the scene by eliminate all the background pixels.

In this step, mixture of Gaussian is used to model the background pixel to adapt the dynamic scene. (Zivkovic, 2004). The pixel is considered foreground when the pixel value does not fit to the distribution of the model. To tackle the slow illumination change due to the change in time of a day (day to afternoon, afternoon to night) or foreground object become a background (car parked into home area), the model will consist only previous 500 frames to model the background by update the model using the new frames and remove the old frames. Sometimes, suddenly illumination might happens, in this case the system will check if the detected foreground pixels are more than 60% of the total pixels of an image frame, then the background model will be reinitialized. (Haritaoglu et al., 2000).

The background model separates the foreground pixels from background pixels and obtains the foreground mask. Foreground mask is a binary image where the background pixel is assigned as 0 (or black) and the foreground pixel is assigned as 255 (or white) as shown in Figure 4.12c). The foreground mask is then undergoing for morphology operation to eliminate some of the noises.

a)

b)

c)

Figure 4.12 Images with original frame (top left), background image (top right), and foreground mask (center bottom)

4.3.1.1 Noise Handling

Noises such as isolated points can be removed by using hit-or-miss transformation. This operation is finding all the pixels' patterns that match exactly with the model. Since the isolated points are required to be removed, therefore the models used for this operation is shown in Figure 4.13.

0	0	0
0	1	0
0	0	0

1	1	1
1	0	1
1	1	1

Figure 4.13 Model used for hit-or-miss transformation which label as D for left mask and W for right mask

With this 2 model, all the isolated points can be filtered out by apply the following equation:

$$(A \ominus D) \cap (A^c \ominus W)$$

Where A is the foreground mask produced by background subtraction and $\ominus$ is erosion morphology operation. The hit-or-miss transformation will produce a white pixel in the center of mask when both of the models are matched. Therefore, by applying intersect operation between foreground mask and negation of the result from hit-or-miss transformation, the isolated points will be removed.

Next, in order to further remove the noises in foreground masks, closing morphology operation is used to fill the gaps found within the single object blob or connects 2 object blobs which refer to the same object. A rectangle 5x5 kernel is used for this closing operation because 3x3 kernel underperforms the hole filling operation due to the gaps between 2 blobs is large and 7x7 kernel is not consider due to the large kernel size might merge the object blobs with other noise pixels. Figure 4.14a) shows the result of closing morphology operation by using the kernel showed in Figure 4.14b).

a)

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

b)

Figure 4.14 Result of closing morphology operation and its kernel

4.3.2 Connected Components Analysis

From the result of foreground segmentation process, connected components analysis is used to extract each of the blob object detected. The connected components analysis is depends on the connectivity defined which is either 8-connectivity or 4-connectivity. The 2 pixels are considered to be in the same blob as long as there is a path from one of the pixel to another based on the connectivity type used. In this project, 8-connectivity is used for the process. Based on the Figure 4.14, there are 2 blobs detected based on the 8-connectivity.

Besides, for each of the blobs detected, sometimes it might be very small that the particular blob provides very less information. In this case, the blob is considered as a noise which is needed to be filtered out. Therefore, the blob is removed if its area is less than 100units^2 as shown in Figure 4.15 where the group of few pixels is removed and 2 blobs are detected (based on Figure 4.14a).

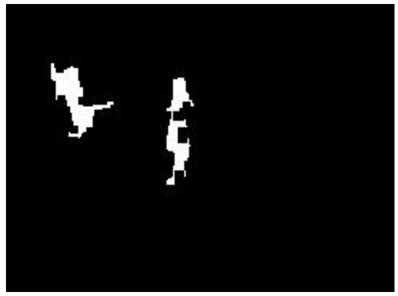


Figure 4.15 Result of connected components analysis and blob filtering

4.3.3 Tracking

Tracking objects that appear in the camera scene over time is an important task to identify robbery scene. When the intruders break into the home area, several assumptions are made as follows:

- Only object from outside area move into inside area are consider intruders
- Object that appear in inside area will never become an intruders

With these assumptions, area classification line (from section 4.1.1) will be used as a reference line for identify intruders.

After all the moving objects segmented out from the scene, object tracking technique will be applied to each of these objects individually. Figure 4.16 shows the flow chart of the proposed object tracking technique. The description of the object tracking technique will be presented from section 4.3.3.1 to section 4.3.3.9.

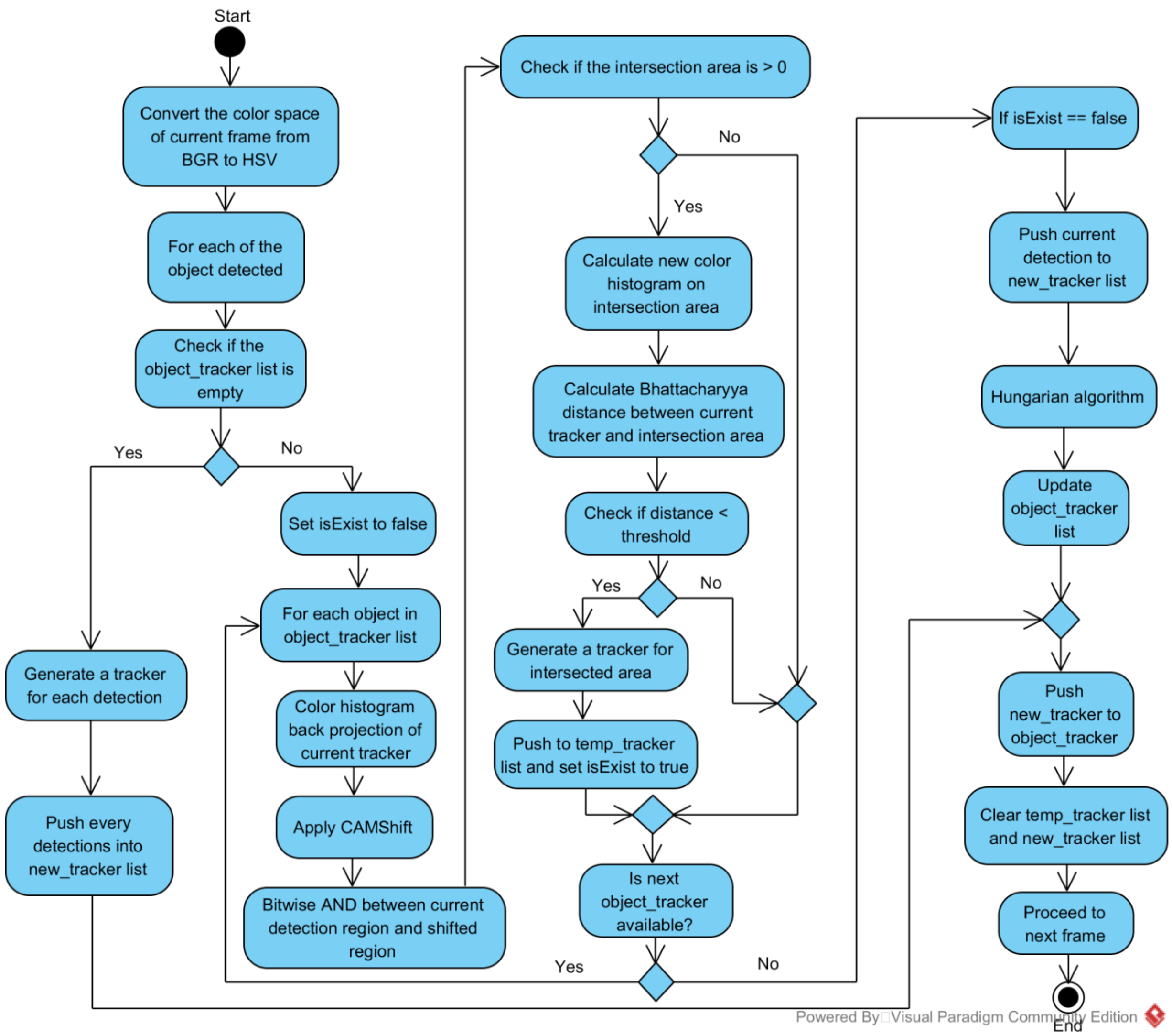


Figure 4.16 Object tracking flow chart

4.3.3.1 Overview

From section 4.3.3.1 to section 4.3.3.9, the detail on the process of object tracking will be discussed. Following terms will be used to identify some of the important variables or data structures in order to clarify the explanation of the technique proposed.

i. object_tracker list – An array of current Tracker
ii. temp_tracker list – An array of temporary generated Tracker
iii. new_tracker list – An array of newly generated Tracker for next frame

4.3.3.2 Object Tracker

In this section, the overall object tracking method is explained and each of the detail sub-process is described from section 4.3.3.3 to section 4.3.3.9. Object Tracker is a data structure or descriptor that defines the characteristics of the each blob objects detected (or detections). It is used to identify each of the object appear in the scene captured. Besides, object tracker also used to determine the object track path by associate the object location in 2 consecutive frames. It consists of several characteristics as follows:

1. HSV color histogram
    - A 3-D color histogram which consists of Hue, Saturation, and Value
2. Area label
    - To determine the tracker is originally from outside area or inside area
3. Warning label
    - To determine the tracker is intruder or not
4. Track path
    - An array that store a list of points that formed a trajectory of the tracker
5. Region
    - To define the respective object region
6. Kalman filter
    - A motion model used to predict the tracker movement

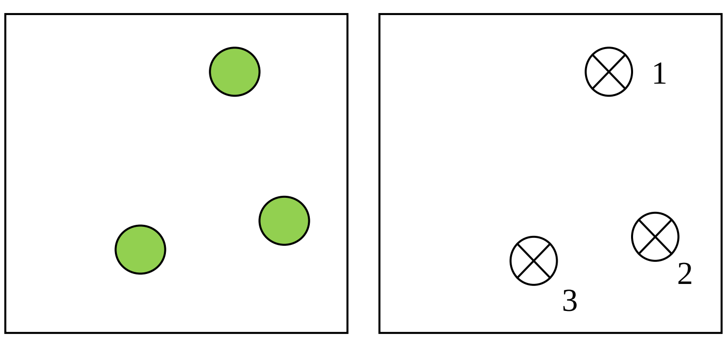
When the all object blobs are detected, each of the detections will generate its tracker by using its color histogram, position, area label, and region. If the object_tracker list is empty in current frame, these generated trackers just added into the object_tracker list at the last step of the object tracking process as shown in Figure 4.16. In next frame, these trackers in object_tracker list will be used to find their respective new state based on the new detections.

In case when the object_tracker list is not empty which means that there are detections appeared in previous frame. Therefore, each of the new trackers generated by the detections in current frame will be pushed into the temp_tracker list. After that, an association is linked between trackers from object_tracker list and trackers from temp_tracker list. Each of the trackers in a list can be assigned to at most one tracker from the other list. If the association is established, the tracker which is assigned is updated by using the information from its assigned tracker. If the association cannot be established, then the unassigned trackers in temp_tracker list will be act as a new tracker to be added into object_tracker list at the end of the process and the unassigned object trackers in object_tracker list will be removed and assumed that it is disappeared in current frame. During the tracker update, the new position of the tracker will be pushed into the track path to update the trajectory of the tracker and other update will be discussed in section 4.3.3.9.

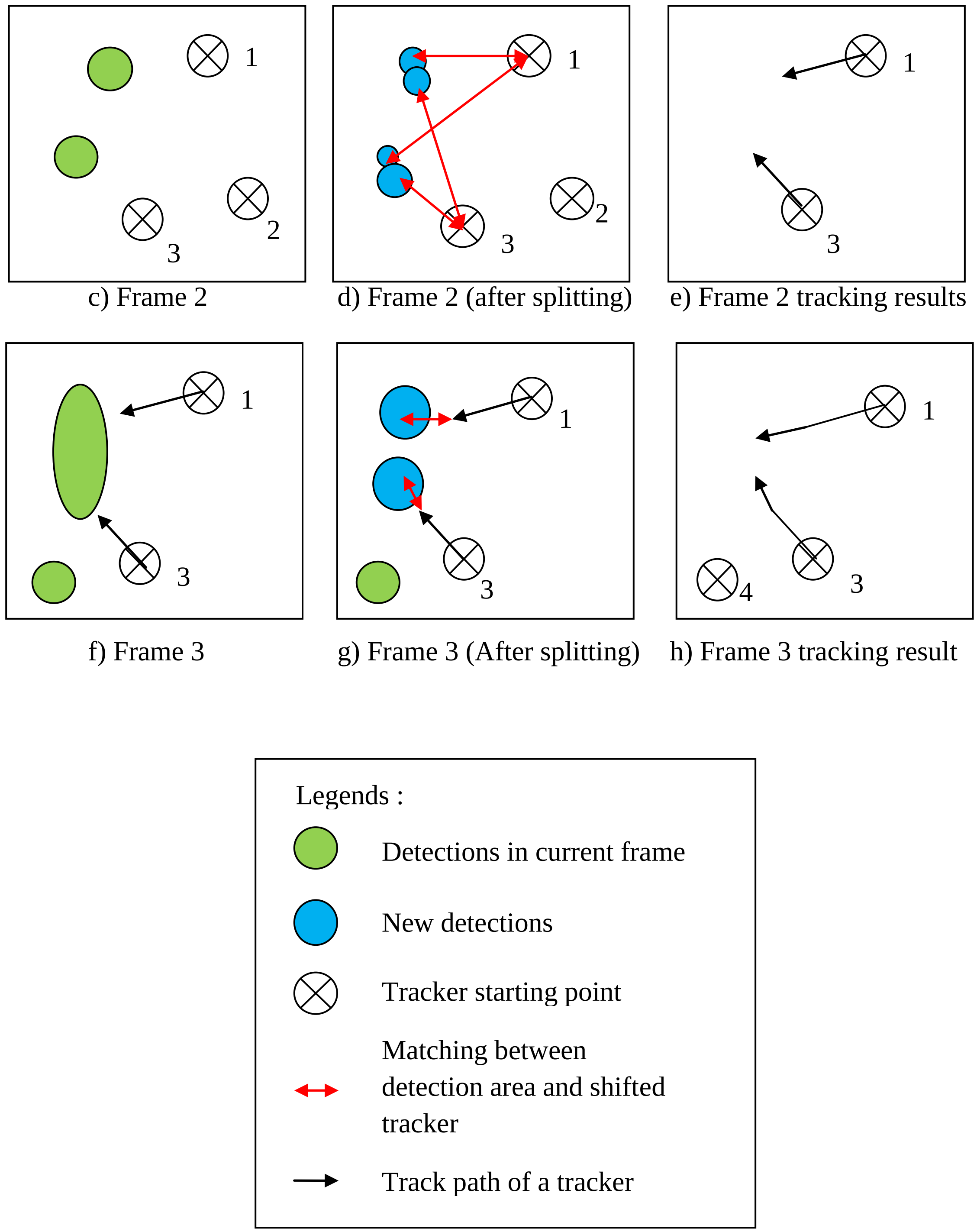
However, by using the mentioned method to perform the object tracking, it will fail to locate the objects when the 2 or more of the tracked objects are merged together. The method will only track one of the 2 trackers and another tracker will be considered as disappeared and hence removed from the object_tracker list.

Therefore, a modification is done on the method when the detections are retrieved and object_tracker list is not empty. After all the detections are extracted, each of the detections will be split into 1 or more new detections by using the information from each trackers in object_tracker list. Firstly, for each of the detections, a CAMSHIFT algorithm will be applied with each of the trackers in object_tracker list on current frame to locate its shifted region. If the shifted region is overlap with the detections region, the HSV color histogram is computed from the overlapping area and then computes the distance between the resultants histogram with the histogram from the trackers. If the distance between resultant histogram and tracker’s histogram is small, then a new detection is formed by using the information from this overlapping area and its respective tracker will be generated and added into the temp_tracker list. Since the original detections might overlapping with 2 or more shifted region generated by different trackers in object_tracker list. As a result, the detection will be split into several new detections and the original detection is removed. At the end of the splitting process, there will be a list of possible temporary generated trackers for trackers in object_tracker list to assign. If the detection cannot be split into new detections due to the high histogram distance with each of the trackers in object_tracker list, the detection will be considered as new object blob that appear to the scene in current frame.

Figure 4.17 illustrates the process mentioned in this section in a sequence of frames. In Figure 4.17a), there are 3 detections available, since the object_tracker list is empty, therefore three of them are initialized as 3 tracker and added into object_tracker list for next frame which shown in Figure 4.17b). In the next frame shown in Figure 4.17c), there are 2 detections and 3 trackers available, now the object_tracker list is not empty, therefore the detections will undergo the splitting process where each of the new detections are formed by intersect the original detection region with shifted region of tracker. In results 4 new detections are formed and the original 2 detections are removed, assuming that there is no intersection area between any detection and shifted region of tracker 2 as shown in Figure 4.17d). After finish the assignment process, the result of the tracker update is shown in Figure 4.17e) and tracker 2 is removed from object_tracker list because it didn't assign to any detections available. In Figure 4.17f), there are 2 detections available and the detection with larger area is assumed to be the merging of 2 detections in previous frame, after undergo splitting process, 2 new detections are formed from the larger detection and the original detection is removed as shown in Figure 4.17g). In Figure 4.17g), the smaller detection is initialized as new object because the color histogram of intersection area between the smaller detection and shifted region of the 2 trackers is not match with the color histogram of the 2 trackers. Finally, Figure 4.17h) is result after the process of assignment and tracker update.



a) Frame 1 b) Frame 1 tracking result



i)

Figure 4.17 Illustration of tracker update process

4.3.3.3 Tracker Initialization

Each of the detections needs to generate its tracker which can describe the detection itself. Some information such as color histogram, position, area label, and region are required to generate a tracker. Color histogram is obtained by using the blob objects as a mask to accumulate all the object's pixels from HSV image. Next, the region of the object is defined as the minimum area of rectangle to cover all the white pixels in the object mask. By obtaining the object region, the position of the object can be easily obtained by locating the centroid of the region. Lastly, the area label is obtained by using the centroid of the object with the equation (2) in section 4.1.1. If the LHS of the resultant equation is greater than zero (> 0), the object is defined as enter the scene in inside area, else the object is located in outside area. Figure 4.18 shows the owner object mask, region and position which are labeled as white pixels, green rectangle, and red cross respectively.

Figure 4.18 Images with original frame (left) and information obtain from the object's blob (right)

4.3.3.4 Continuously Adaptive Mean Shift

Each of the detections will be split into several new detections by using CAMSHIFT algorithm which the current tracker's color histogram is used as feature space for CAMSHIFT algorithm. (Bradski, 1998).

Since CAMSHIFT algorithm required to shift the current region to a more confidence region, color histogram back projection is applied on the current frame to obtain a confidence map (or probability image). Back projection is a process of measure how well the pixels of a given image fit the distribution of color histogram. Figure 4.19 shows the results of back projection with 2 trackers from which are the intruders and the owner. The results of back projection shows that the brighter the region, the more confidence for that region to fit the color histogram. This step is repeated for every tracker in the object_tracker list.

Figure 4.19 Image with original frame (left), back projection of the intruders' tracker (center), and back projection of the owner's tracker (right)

After that, CAMSHIFT algorithm is used to shift the current region to a brighter region based on the confidence map generated by back projection. The CAMSHIFT algorithm can locate the new detection region through use of mean-shift technique and adapt the changes in object size provided that the initial starting region. The initial region is input as tracker region in this step. Figure 4.20 shows the result of CAMSHIFT algorithm where the red box represents the tracker region and the green box represents the shifted region.

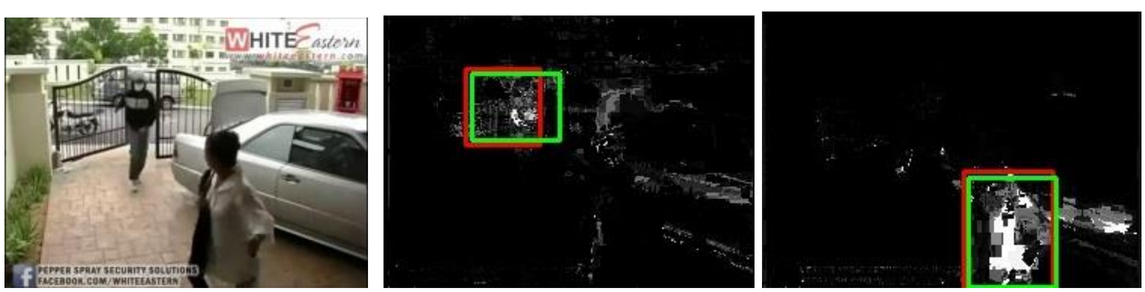
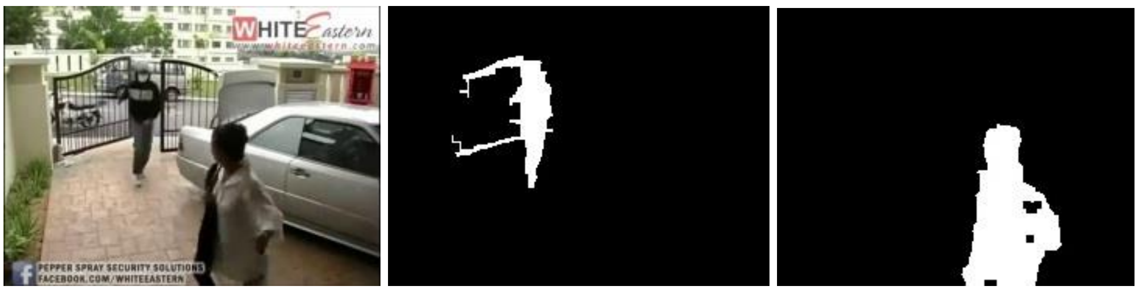


Figure 4.20 Image with original frame (left), CAMSHIFT of the intruders' tracker (center), and CAMSHIFT of the owner's tracker (right)

Once the shifted region is obtained, a bitwise AND operation is used to locate the new detections based on the original detections. For each combination between the all the detections and all the shifted regions, the intersection area is found by using bitwise AND operation as shown in Figure 4.21. Figure 4.21 illustrate the process of the formation of new detection from the original detections and shifted region. In Figure 4.21f, there is no overlapping region, therefore no detections is formed. When a detection has no overlapping region with all the shifted regions, the detection is considered a new object that appeared to the scene.



a) Original frame

b) Original detections for intruders

c) Original detections for owner

d) New detection between intruders' shifted region and b)

e) New detection between owners shifted region and c)

f) No detection found between owners shifted region and b) / intruders' shifted region and c)

Figure 4.21 Illustration of the formation of new detections

4.3.3.5 Bhattacharyya Distance

Next, in order to determine whether the new detection formed is reliable, a color histogram is generated based on the new detection mask and compare with the respective tracker's color histogram. In this process, Bhattacharyya distance is used to measure the distance between two histograms. The output of Bhattacharyya distance is from 0 to 1 where 0 shows that the given two histograms are identical whereas 1 represents that the given two histograms are completely different. If the distance is less than 0.7, the new detection is acceptable for further process by added to temp_tracker list, else it will be ignored. The high threshold 0.7 is used because in this step, the more data generated the more possible accurate result can be obtained in later stage since the information is sufficient enough. If all the new detections formed from current detection are having a Bhattacharyya distance more than 0.7, the detection is consider a new detection where its tracker is generated and pushed into the new_tracker list at the end of the process.

4.3.3.6 Affinity Model

During the assignment process, a measurement is required to measure how similar for the given 2 trackers in order to make optimal assignments between all the trackers in temp_tracker list and object_tracker list. Therefore, affinity model is introduced to tackle this problem. The affinity model will results a value which determine the degree of similarity between 2 trackers where 0 represents the 2 trackers are completely different and 1 represents that the 2 trackers are exactly the same. The affinity measurement considers 3 characteristics of the tracker which are color histogram, position, and motion. The affinity score between 2 trackers is the product of these 3 affinity model.

Color Affinity

In order to measure the affinity between two color histogram, Bhattacharyya distance is used where the distance is measure from 0 to 1 which interpreted as from exactly the same to completely different respectively. Since affinity measure how similar given 2 histograms, therefore the Bhattacharyya distance obtained is required to subtract from 1 to obtain color affinity.

Position Affinity

For position affinity, the absolute difference between previous position of tracker in object_tracker list and current position of tracker in temp_tracker list. Below equation shows the position affinity measured in exponential of negative distance.

$$\Lambda(P_1, P_2) = e^{-\left(\frac{abs(x_1 - x_2)}{x_1 + x_2} + \frac{abs(y_1 - y_2)}{y_1 + y_2}\right)}; P_1(x_1, y_1), P_2(x_2, y_2) \quad (4)$$

Motion Affinity

Lastly, the motion affinity is measured with equation (4) but with different points where the points are the previous predicted position of tracker in object_tracker list based on velocity and current position of tracker in temp_tracker list. The predicted position of a tracker is obtained through the process of Kalman filter. (Welch and Bishop, 1995). When an object first appears to the scene, an instance of Kalman filter is assigned to it along with tracker. Each objects' Kalman filter will correct and update its measurements (position and velocity) when the object tracker is updated with assigned tracker's information. At the start of tracking process in every frame, a prediction has to be done to obtain the predicted position of each of the trackers in object_tracker list based on the corrected measurements.

4.3.3.7 Tracker Confidence

Tracker confidence provides a measure of degree on how reliable on the respective tracker. After the temp_tracker list is generated, each of the tracker in object_tracker list will be undergo a classification where the tracker with track path length less than 20 is consider having low confidence else the tracker is consider having high confidence. The track path length is defined as the number of frame of this tracker exists since it appears to the scene. The purpose of the classification is to set a priority to the trackers where the high confidence trackers have the high priority to undergo the assignment step first before the low confidence trackers. Once the high confidence tracker is assigned, an association is established between one of the trackers from temp_tracker list and one of the trackers from object_tracker list. The assigned tracker will be removed from their respective list. In case when the high confidence tracker is not assigned to any tracker from temp_tracker list, it will be removed from the list as well. After that, the temp_tracker list will be used for the same assignment process with low confidence tracker. Lastly, all the remaining trackers from both tracker lists that didn't assign to any other tracker will be removed.

4.3.3.8 Hungarian Assignment Algorithm

In the assignment step where associations are need to be established between the trackers in temp_tracker list and object_tracker list. This step require to generate a list of pairs between trackers from temp_tracker list and object_tracker list where the total affinity score for the all the pairs are optimal whereas the unassigned tracker will be eliminated. Therefore, from temp_tracker list and object_tracker list, a profit matrix is formed by making object_tracker list and temp_tracker list as columns and rows respectively. Each cell in the profit matrix is representing the affinity score, between the tracker from temp_tracker list and tracker from object_tracker list.

By using the matrix generated, Hungarian algorithm is used to find out the optimal assignment between these 2 lists as illustrated in Figure 4.22. (Kuhn, 1955). Since the Hungarian algorithm is finding the overall minimal cost which is contradict with the affinity score defined in section 4.3.3.6. In order to modify the matrix generated to cost matrix, the matrix is normalized to 0 to 255 and then subtract each cells from 255 to obtain cost matrix. With the aforementioned algorithm, it can only solve the assignment problem when the number of row is equal to the number of column. In this project, the number of tracker in temp_tracker list and number of tracker in object_tracker list are usually not equal to each other. Therefore, a Munkre's variant of Hungarian assignment algorithm is used. (Bourgeois and Lassalle, 1971). The Munkre's assignment algorithm is able to obtain the optimal assignment in square matrix as well as in rectangular matrix with the constraint that the number of column must be at least as much as the number of row. The aforementioned constraint can be solved easily by adding a zero-column to the matrix until the number of column is more than or equal to number of row. The algorithm performs by only following the 7-steps as shown below to find the optimal solution to the assignment problem.

1. Search the smallest element in each row and then subtract it from every element in its respective row. Next, go to step 2.
2. For each of the elements in the resulting matrix, if the element is zero valued and there is no starred in its row or column, star this zero (Z) element by star its row and column. Next, go to step 3.

3. Cover each column containing a starred zero. If the number of columns covered is equal to the number of column of the original column, the starred zeros describe a complete set of unique assignments. In this case, go to step 7, else, go to step 4.
4. Search for an uncovered zero and set it as prime. If there is no starred zero been found in the row containing this primed zero, go to step 5. Else, cover this row and uncover the column containing the zero that labelled as starred. Proceed in this way until all the zeros are covered. Save the smallest uncovered value and go to step 6.
5. In this step, a series of alternating primed and starred zeros are generated as follows. Let Z_0 indicate the uncovered primed zero found in step 4. Let Z_1 represent the starred zero in the column of Z_0 (if any). Then Z_2 denote the primed zero in the row of Z_1. Proceed until the series terminates with a primed zero that has no starred zero in its column. Then, the star label for each starred zero of the series is removed, followed by labelling star for each primed zero of the series, then erase all primes and uncover every line in the matrix. Lastly, return to step 3.
6. The smallest value found in step 4 is added to every element of each covered row, and subtract the smallest value found from every element of each uncovered column. After that, return to step 4 without update any stars, primes, or covered lines.
7. Assignment pairs are indicated by the positions of the starred zeros in the resultant matrix where the starred zeros are represented as 1 in assigned matrix.

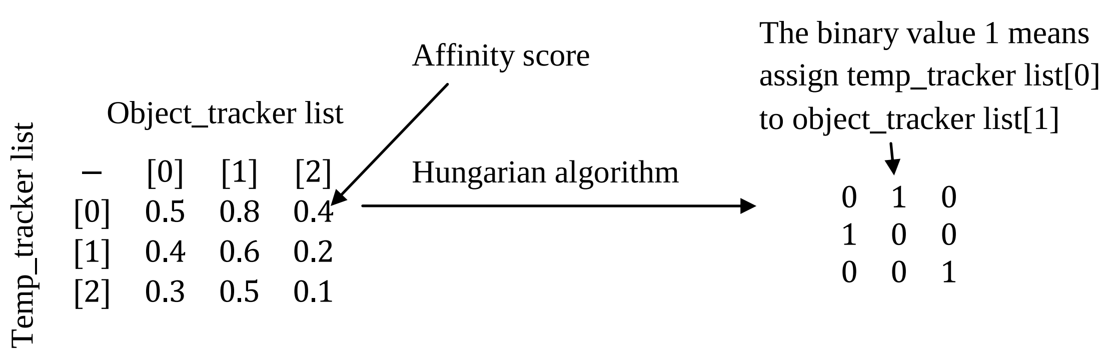


Figure 4.22 Result produced by using Hungarian algorithm

Although the result produced by Hungarian algorithm is optimal, but there is some case that the affinity score for a particular cell is very low and it is selected as the optimal assignment as shown in the last column of the matrix in Figure 4.22. In this case, a threshold is required to determine whether the assignment is reliable. When the affinity score is too low, the particular cell is considered not reliable and hence in the result matrix where the binary value will be set from 1 to 0. In others word, the tracker in the object_tracker list is assume to be disappeared in current frame since it cannot pairs with any of the trackers in temp_tracker list.

4.3.3.9 Tracker Update

When the tracker in object_tracker list has been assigned to a tracker from temp_tracker list, then the tracker in object_tracker list is required to update by using information from the assigned tracker in temp_tracker list. This means that the assigned tracker in temp_tracker list is the new state for the tracker in object_tracker list. The update process involves color histogram, position, Kalman filter, track path, region, and warning label.

The color histogram is updated by replacing the color histogram of the tracker in object_tracker list with the color histogram of the assigned tracker. Then, the position, Kalman filter, and track path will be updated by using the position information from the assigned tracker. During the update on the Kalman filter, the new position is used as measurements to correct the Kalman filter state so that the prediction in the next frame is more reliable. At the same time, the track path is updated by push the new position into the list. Next, the region is updated by replacing the region information of tracker in object_tracker list with the assigned tracker. After that, the update process on the warning label is by identifying the area located for the tracker in temp_tracker list by using the equation (2) in section 4.1.1. If the area label of the tracker in temp_tracker list is “inside” and the area label for the tracker in object_tracker list is “outside”, then the warning label of this tracker is set as true. Lastly, all the trackers in temp_tracker list will be removed once the update is finish.

4.4 Intrusion Detection

The level of security of the proposed system is designed in 4 phases which are safe, alert, warning, and danger. In safe phase, there is no intruder detected and the gate is detected as closed. Next, the alert phase is entered when the gate is detected as opened under the system monitor but no intruder is detected. When there is intruder detected and the gate is closed then the system is entering the warning phase. Lastly, the danger phase is entered whenever there is an intrusion action detected.

4.5 Alarm System

In terms of the alarm system, the alarm system will behave differently which depends on the phase of the security. The alarm system does nothing when the security phase is in safe phase. When the security phase reached to alert or warning phase, a low frequency alert sound is produced. Lastly, in danger phase, a high frequency alert sound is produced.

CHAPTER 5 EXPERIMENTAL RESULTS AND DISCUSSION

The experiments are done in 2 aspects, which are gate analysis performance and object tracking performance. The number of scenarios which used for the experiments are 5 where each of the scenarios are shown in sample frame in Figure 5.1.

a) Scenario 1

b) Scenario 2

c) Scenario 3

d) Scenario 4

e) Scenario 5

Figure 5.1 Sample frame of scenarios that used as experiments

5.1 Gate Analysis Performance

The experiments in gate analysis are conducted by observing results generated by the system.

5.1.1 Experiment on scenario 1

In the first scenario, the results are obtained as shown in Figure 5.2.

a)

b)

c)

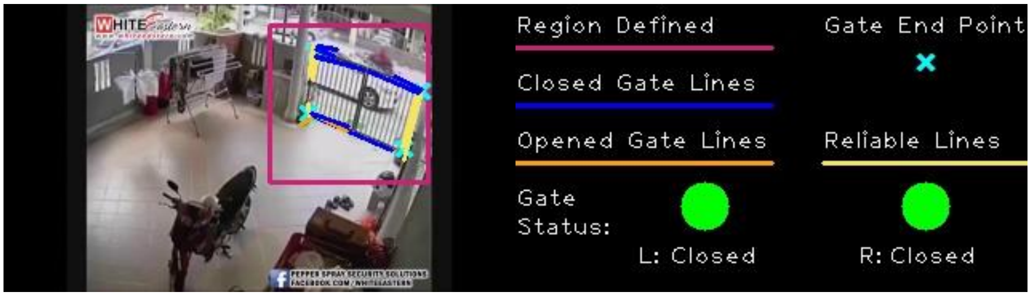
d)

Figure 5.2 Experiment results of gate analysis obtained from scenario 1

Based on Figure 5.2, the results showed that performance of gate detection is good when the gate pattern can be seen obviously. The status of the left-side gate can be obtained and display correctly. When the reliable gate line is missing as shown in right-side gate of Figure 5.2c) and Figure 5.2d), the system will remain the latest gate status which is closed. In case where the right-side gate is opened when the reliable gate lines is missing, the gate status will not update which is the drawback of this system since the gate information cannot be obtained when vision towards it is blocked. The solution for this problem might be install another camera in different place so that the system allows to collect information from different camera and then later combine to perform an analysis.

5.1.2 Experiment on scenario 2

In scenario 2, the gate is opened by sliding to inside area instead of outside area. Therefore, following results has been recorded as shown in Figure 5.3.



a)

b)

c)

d)

Figure 5.3 Experiment results of gate analysis obtained from scenario 2

In Figure 5.3, the gate is opened by intruders by sliding the gate towards inside area. The results show that the system is able to detect the orientation of the gate despite how the gate is opened.

5.1.3 Experiment on scenario 3

In scenario 3, the special type gate is used where the gate is divided into several parts where each part can be opened or closed. In this case, the 4 gate points input from the user is not enough to describe the gate as shown in Figure 5.4.

a)

b)

Figure 5.4 Experiment results of gate analysis obtained from scenario 3

The system fails to obtain all the information from the gate due to the gate has many leaves which are 4 in this case. The user can select 4 points that correspond to the opened leaf and left other leaf in closed as shown in Figure 5.5.

a)

b)

Figure 5.5 Experiment results of gate analysis obtained from scenario 3 by changing 4 gate points

As shown in Figure 5.5 the proposed system is not able to get all the gate line due to the design of the gate where the present of horizontal skeleton in top and bottom of the gate leaf which block the vision of the gate leaf. In summary, the system would not able to retrieve the gate information when the gate design is unique.

5.1.4 Experiment on scenario 4

In scenario 4, the surveillance camera is not placed properly such that the camera cannot capture the whole gate region. The results of the gate analysis on scenario 4 are shown in Figure 5.6.

a)

b)

Figure 5.6 Experiment results of gate analysis obtained from scenario 4

From Figure 5.6, the results shows that the system is fail to obtain most of the gate lines due to the improper placement of the camera and poor lighting condition. In summary, the system is not able to provide accurate results when the information provided is very less.

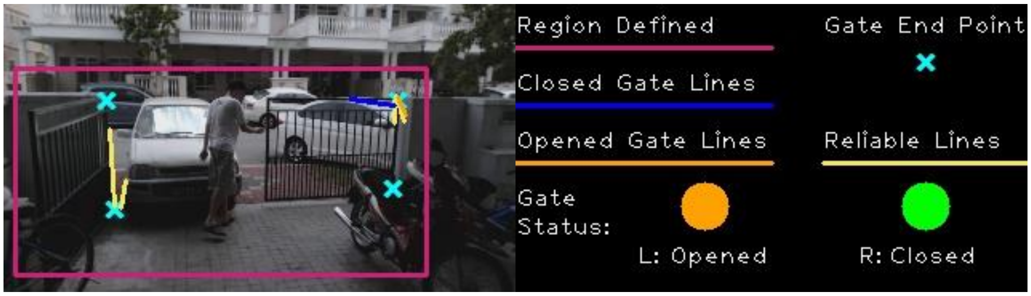
5.1.5 Experiment on scenario 5

In scenario 5, the gate pattern to observe due to the cloudy weather and complex scene. The results of gate analysis are shown in Figure 5.7.

a)

b)

c)



d)

e)

Figure 5.7 Experiment results of gate analysis obtained from scenario 5

Based on the results from scenario 5, the system fail to obtain gate pattern from the scene due to the cloudy weather and the shadow projected to the gate which make the edge of the gate color merge with the shadow color. This problem greatly affects the Canny edge detector to fail in obtaining the gate lines as shown in Figure 5.8. Moreover, Figure 5.7a shows the false detection on gate lines on the left-side gate. This shows that there might be have an object in a scene where the Canny edge results is having similar feature as gate.

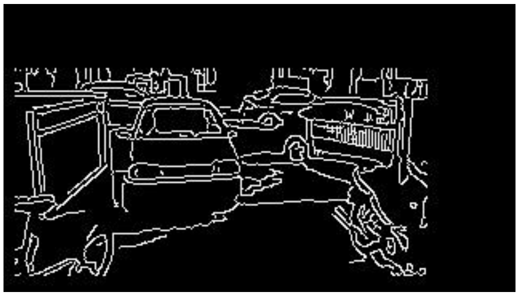


Figure 5.8 Results of Canny edge detector on scenario 5 scene

5.2 Object Tracking Performance

The performance of the object tracking is evaluated by observing the trajectory produced by the system.

5.2.1 Experiment on scenario 1

In scenario 1, the performance of the proposed tracking method is good when the object have unique color in a scene. Figure 5.9 shows the results of object tracking in scenario 1.

a)

b)

c)

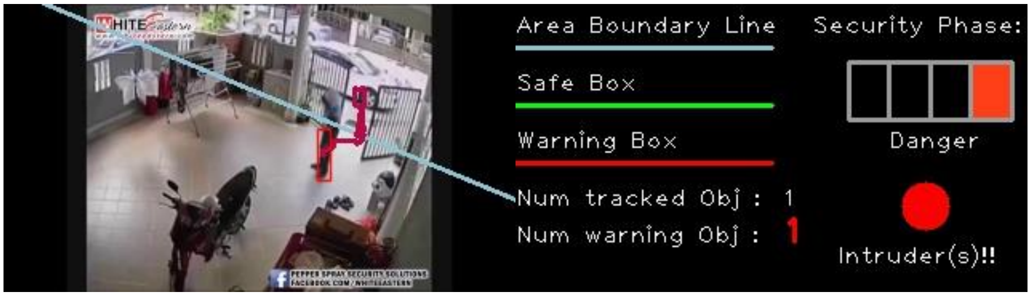
d)

Figure 5.9 Experiment results of object tracking in scenario 1

Figure 5.9 shows object tracking results in intrusion case where the 2 intruders break-into the home and snatch the owner handbag. The results show that the tracking is able to locate the intruders correct until they split up into 2. When the intruders is split up into 2 groups, the tracker only able to locate one of them and the other intruder is assume to be disappear from the scene. For the owner's tracker, the tracker is able to track correctly in Figure 5.9a) and later on the tracker lost its object and a new tracker are reinitialized to tracker the owner as shown in Figure 5.9b). In Figure 5.9d), the owner's tracker is slowly shifted to the intruders when the intruders try to escape from the owner house.

5.2.2 Experiment on scenario 2

Scenario 2 involves multiple intruder break-into the home area with occluding each other as shown in Figure 5.10.



a)

b)

c)

d)

Figure 5.10 Experiment results of object tracking in scenario 2

In scenario 2, there are total 4 intruders break into the home area one by one. The results show that the proposed system able to detect intrusion case 3 out of 4. Besides, in Figure 5.10b), there is a small area is being tracked due to the suddenly illumination change. Usually, these false detections will be disappeared after few frames due to their color distribution are closer and closer to the background.

5.2.3 Experiment on scenario 3

Scenario 3 involves an intruder that attempt to intrude the house twice and both times the intruder also cross the area classification line as shown in Figure 5.11.

a)

b)

c)

d)

Figure 5.11 Experiment results of object tracking in scenario 3

Based on the results from Figure 5.11, both intrusion attempts from the intruder are detected successfully. Besides, other accomplice also tracked successfully in outside area.

5.2.4 Experiment on scenario 4

In Scenario 4, the scene is captured under the poor light condition and the results are shown in Figure 5.12.

a)

b)

c)

Figure 5.12 Experiment results of object tracking in scenario 4

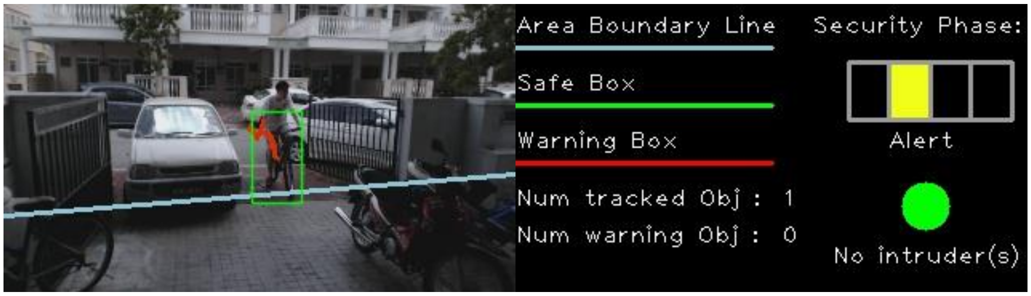
From Figure 5.12, the results show that the system has been performing less accurate due to the poor lighting condition. In Figure 5.12a), the owner's shadow is tracked instead of the real object. This is due to the CAMSHIFT algorithm shift the region to darker area due to the bottom half of the owner body is also dark due to the projection of shadow. In Figure 5.12b), when the first intruder break-in only the owner object is tracked but not the intruder. This result shows that the tracker is not able to locate the intruder when the intruder first appear to the scene due to the less color information obtain from the scene. With this cause, in the following frames, there will be one less tracker to track the intruder. However, with all false detections on previous frames, the system able to detect the second intrusion successfully as shown in Figure 5.12c).

5.2.5 Experiment on scenario 5

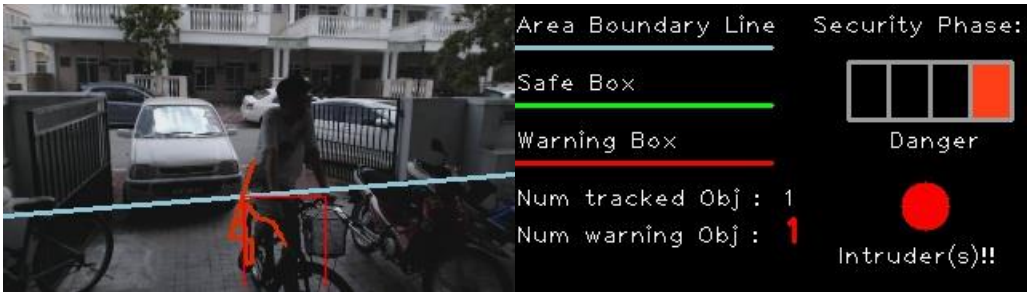
In scenario 5, the intruder is riding a vehicle (such as bicycle, motorcycle, and car) to break-into the home area as shown in Figure 5.13.

a)

b)



c)



d)

Figure 5.13 Experiment results of object tracking in scenario 5

Scenario 5 shows that the background model is able to handle the slow illumination change on the scene as shown in Figure 5.13a) and Figure 5.13b). During this period, no false detection on moving object is occurred. In Figure 5.13c) and Figure 5.13d), the tracker is not able to track the intruder for a single frame. As a result, the tracker is initialized again to track the intruder. This shows that the tracker will fail when the tracker cannot locate its new location for a single frame which is due to the mismatch between shifted region and tracker.

CHAPTER 6 CONCLUSION

In this paper, the crime rate in Malaysia is studied and there is an increase in the crime rate in Malaysia. Hence, there is a need for surveillance system to overcome the problems. The proposed surveillance system can help users to monitor their home area semi-automatically to ensure their home area is monitored.

This project provides a robbery scene detection surveillance system that specifically contributes to the home intrusion detection. The system can detect the intrusion case by using computer vision techniques. The system will monitor and analyze the video frames captured and detect whether the intrusion has happened.

The following techniques have been developed in this project which are area classification, gate analysis, and object tracking. Area classification provides a boundary line that can separate a scene into two areas which are outside area and inside area. By using the classification line, the tracked object can be detected as intruder when the particular object is moving from outside area to inside area. To further improve the accuracy of the system, gate analysis technique is used to provide some evidences to the system in order to cover some of the false detections.

Throughout this project, several problems have been encountered. Firstly, the scope of the project is very wide such that with a limited time provided, it is very difficult to study all the techniques available in research papers. Secondly, the scenario for robbery scene can be varied in type of gate, method of robbery, number of intruders, placement of camera, surrounding light, weather, and more. Some of these can be controlled while the others cannot be controlled. This further increases the problems for researcher to implement a perfect surveillance system.

For future works, the proposed system can be improved by using a more robust model to describe all the object appeared to the scene which can handle splitting, merging, and occlusion between objects. Next, the gate analysis can be improved by include other type of gate instead of only applicable to two leaves swing gate. On the other hand, this project can be extended by include some action recognition/classification and robbery event behaviour analysis to detect whether the intruder is perform some harmful action such as killing, stealing, threatening, and more. Moreover, some evidences other than gate status such as screaming sound, presents of blood, and presents of weapons can be used to further improve the system. Lastly, a human detector can be used in replace to the moving object segmentation.

BIBLIOGRAPHY

Avidan, S. (2004) ‘Support vector tracking’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8), pp. 1064 – 1072.

Bar-Shalom, Y. and Fortmann, T. E. (1988) *Tracking and Data Association*. Academic Press.

Bertalmio, M., Sapiro, G. and Randall, G. (2000) ‘Morphing active contours’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7), pp. 733–737.

Black, M. J. and Jepson, A. D. (1998) ‘EigenTracking: Robust matching and tracking of articulated objects using a view-based representation’, *International Journal of Computer Vision*, 26(1), pp. 63–84.

Blank, M., Gorelick, L., Shechtman, E., Irani, M. and Basri, R. (2005) ‘Actions as space-time shapes’, *Tenth IEEE International Conference on Computer Vision*, 2, pp. 1395–1402.

Bodor, R., Jackson, B. and Papanikolopoulos, N. (2003) ‘Vision-Based Human Tracking and Activity Recognition’, *In Proceedings of the 11th Mediterranean Conference on Control and Automation*, 1, pp. 18–20.

Bourgeois, F. and Lassalle, J.-C. (1971) ‘An extension of the Munkres algorithm for the assignment problem to rectangular matrices’, *Communications of the ACM*, 14(12), pp. 802–804.

Bradski, G. R. (1998) 'Real time face and object tracking as a component of a perceptual user interface', *Proceedings Fourth IEEE Workshop on Applications of Computer Vision*, pp. 214–219.

Brand, M., Oliver, N. and Pentland, A. (1997) 'Coupled hidden Markov models for complex action recognition', *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 994–999.

Brendel, W. and Todorovic, S. (2009) 'Video object segmentation by tracking regions', *2009 IEEE 12th International Conference on Computer Vision*, pp. 833–840.

Broida, T. J. and Chellappa, R. (1986) 'Estimation of Object Motion Parameters from Noisy Images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1), pp. 90–99.

Canny, J. (1986) 'A Computational Approach to Edge Detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), pp. 679–698.

Comaniciu, D., Ramesh, V. and Meer, P. (2003) 'Kernel-based object tracking', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), pp. 564–577.

Dalal, N. and Triggs, B. (2005) 'Histograms of Oriented Gradients for Human Detection', *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, pp. 886–893.

Du, Y., Chen, F. and Xu, W. (2007) 'Human interaction representation and recognition through motion decomposition', *IEEE Signal Processing Letters*, 14(12), pp. 952–955.

Fiaz, M. K. and Ijaz, B. (2010) 'Vision based human activity tracking using artificial neural networks', *2010 International Conference on Intelligent and Advanced Systems*, pp. 1–5.

Gorelick, L., Blank, M., Shechtman, E., Irani, M. and Basri, R. (2007) 'Actions as Space-Time Shapes', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12), pp. 2247–2253.

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.

Hausdorff, F. (2005) *Set theory*. Providence, Rhode Island: AMS Chelsea Publishing, American Mathematical Society.

Horprasert, T., Harwood, D. and Davis, L. S. (1999) 'A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection', *IEEE International Conference on Computer Vision*, 99, pp. 1–19.

Huttenlocher, D. P., Noh, J. J. and Rucklidge, W. J. (1993) 'Tracking non-rigid objects in complex scenes', *1993 Proceedings 4th International Conference on Computer Vision*, pp. 93–101.

Isard, M. and Blake, A. (1998) 'Condensation - conditional density propagation for visual tracking', *International Journal of Computer Vision*, 29(1), pp. 5–28.

Jain, A. K., Duin, R. P. W. and Mao, J. (2000) 'Statistical pattern recognition: A review', *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(1), pp. 4–37.

Jepson, A. D., Fleet, D. J. and El-Maraghi, T. R. (2003) 'Robust online appearance models for visual tracking', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), pp. 1296–1311.

Kailath, T. (1967) 'The divergence and Bhattacharyya distance measures in signal selection', *IEEE Transactions on Communication Technology*, 15(1), pp. 52–60.

Ke, S.-R., Thuc, H., Lee, Y.-J., Hwang, J.-N., Yoo, J.-H. and Choi, K.-H. (2013) 'A Review on Video-Based Human Activity Recognition', *Computers*, 2(2), pp. 88–131.

Kitagawa, G. (1987) 'Non-Gaussian State Space Modeling of Nonstationary Time Series', *Journal of the American Statistical Association*, 82(400), pp. 1032–1063.

Koller, D., Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B. and Russell, S. (1994) ‘Towards robust automatic traffic scene analysis in real-time’, *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, pp. 126–131.

Kuhn, H. W. (1955) ‘The Hungarian method for the assignment problem’, *Naval Research Logistics Quarterly*, 2(1-2), pp. 83–97.

Lao, W., Han, J. and De With, P. H. N. (2009) ‘Automatic video-based human motion analyzer for consumer surveillance system’, *IEEE Transactions on Consumer Electronics*, 55(2), pp. 591–598.

Lowe, D. G. (1999) ‘Object recognition from local scale-invariant features’, *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, pp. 1150–1157.

Lowe, D. G. (2004) ‘Distinctive Image Features from Scale-Invariant Keypoints’, *International Journal of Computer Vision*, 60(2), pp. 91–110.

Monnet, A., Mittal, A., Paragios, N. and Ramesh, V. (2003) ‘Background modeling and subtraction of dynamic scenes’, *Proceedings of IEEE International Conference on Computer Vision*, pp. 1305–1312.

Murphy, K. P. (2002) *Dynamic bayesian networks: representation, inference and learning*. PHD Dissertation thesis. University of California, Berkeley. .

Oliver, N., Rosario, B. and Pentland, A. (2000) ‘A Bayesian Computer Vision System for Modeling Human Interactions’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 831–843.

Overseas Security Advisory Council (2015) *Malaysia 2015 Crime and Safety Report*. Available at: https://www.osac.gov/pages/ContentReportDetails.aspx?cid=17215 (Accessed: 29 August 2015).

Permuter, H., Francos, J. and Jermyn, I. (2006) ‘A study of Gaussian mixture models of color and texture features for image classification and segmentation’, *Pattern Recognition*, 39(4), pp. 695–706.

Piccardi, M. (2004) ‘Background subtraction techniques: a review’, *2004 IEEE International Conference on Systems, Man and Cybernetics*, 4, pp. 3099–3104.

Poppe, R. (2010) ‘A survey on vision-based human action recognition’, *Image and Vision Computing*, 28(6), pp. 976–990.

Rabiner, L. R. (1989) ‘A tutorial on hidden Markov models and selected applications in speech recognition’, *Proceedings of the IEEE*, 77(2), pp. 257–286.

Rabiner, L. and Juang, B. (1986) ‘An introduction to hidden Markov models’, *IEEE Acoustics, Speech, and Signal Processing Magazine*, 3, pp. 4–16.

Rangarajan, K. and Shah, M. (1991) ‘Establishing motion correspondence’, *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 103–108.

Reid, D. (1979) ‘An algorithm for tracking multiple targets’, *IEEE Transactions on Automatic Control*, 24(6), pp. 843–854.

Sato, K. and Aggarwal, J. K. (2004) ‘Temporal spatio-velocity transform and its application to tracking and interaction’, *Computer Vision and Image Understanding*, 96(2), pp. 100–128.

Schuldt, C., Laptev, I. and Caputo, B. (2004) ‘Recognizing human actions: a local SVM approach’, *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, 3, pp. 32–36.

Sempena, S., Maulidevi, N. U. and Aryan, P. R. (2011) ‘Human action recognition using Dynamic Time Warping’, *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pp. 1–5.

Sethi, I. K. and Jain, R. (1987) ‘Finding Trajectories of feature points in a Monocular image sequence’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1), pp. 56–73.

Stauffer, C. and Grimson, W. E. L. (1999) ‘Adaptive background mixture models for real-time tracking’, Proceedings. *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246–252.

Stauffer, C. and Grimson, W. E. L. (2000) ‘Learning patterns of activity using real-time tracking’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 747–757.

Sullivan, J. and Carlsson, S. (2002) ‘Recognizing and tracking human action’, *Proceedings of the 7th European Conference on Computer Vision*, pp. 629–644.

Tao, H., Sawhney, H. S. and Kumar, R. (2002) ‘Object tracking with Bayesian estimation of dynamic layer representations’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), pp. 75–89.

Terzopoulos, D. and Szeliski, R. (1992) ‘Tracking using Kalman snakes’, *Active Vision*, MIT Press, pp. 3–20.

Veenman, C. J., Reinders, M. J. T. and Backer, E. (2001) ‘Resolving motion correspondence for densely moving points’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1), pp. 54–72.

Veeraraghavan, A., Roy-Chowdhury, A. K. and Chellappa, R. (2005) ‘Matching shape sequences in video with applications in human movement analysis’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12), pp. 1896–1909.

Welch, G. and Bishop, G. (1995) *An Introduction to the Kalman Filter*. North Carolina, USA: University of North Carolina at Chapel Hill.

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.

Yilmaz, A., Javed, O. and Shah, M. (2006) ‘Object Tracking: A Survey’, *ACM Computing Surveys*, 38(4), pp. 1–45.

Yilmaz, A., Li, X. and Shah, M. (2004) ‘Contour-based object tracking with occlusion handling in video acquired using mobile cameras’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), pp. 1531–1536.

Yu, T., Zhang, C., Cohen, M., Rui, Y. and Wu, Y. (2007) ‘Monocular Video Foreground/Background Segmentation by Tracking Spatial-Color Gaussian Mixture Models’, *2007 IEEE Workshop on Motion and Video Computing*, p. 5.

Zhong, J. and Sclaroff, S. (2003) ‘Segmenting foreground objects from a dynamic textured background via a robust kalman filter’, *Proceedings of IEEE International Conference on Computer Vision*, pp. 44–50.

Zivkovic, Z. (2004) ‘Improved adaptive Gaussian mixture model for background subtraction’, *Proceedings of the 17th International Conference on Pattern Recognition, 2004*, 2, pp. 28–31.

APPENDICES A TURNITIN RESULTS

Assignment Inbox: FYP1601

	Info	Dates	Similarity	
FYP report	ⓘ	Start 01-Apr-2016 3:45PM Due 06-May-2016 11:59PM Post 09-Apr-2016 12:00AM	11%	Resubmit View

Originality GradeMark PeerMark

ROBBERY SCENE DETECTION: DETECT INTRUDERS TO HOME AREA

BY CHIN WAI KIT

turnitin 11% SIMILAR -- OUT OF 0

turnitin Originality Report

Processed on: 13-Apr-2016 06:33 MYT
ID: 658509803
Word Count: 20485
Submitted: 1

Document Viewer

ROBBERY SCENE DETECTION: DETECT INTRUDERS TO ...
By Chin Wai Kit

Similarity Index	Similarity by Source	
11%	Internet Sources:	7%
	Publications:	7%
	Student Papers:	5%

turnitin **Turnitin Originality Report**

ROBBERY SCENE DETECTION: DETECT INTRUDERS TO HOME AREA by Chin Wai Kit

From FYP report (FYP1601)

Processed on 13-Apr-2016 06:33 MYT
ID: 658509803
Word Count: 20485

Similarity Index	Similarity by Source	
11%	Internet Sources:	7%
	Publications:	7%
	Student Papers:	5%

Turnitin Originality Report

Processed on: 13-Apr-2016 06:33 MYT
ID: 658509803
Word Count: 20485
Submitted: 1

ROBBERY SCENE DETECTION: DETECT INTRUDERS TO ... By Chin Wai Kit

Similarity Index	Similarity by Source	
11%	Internet Sources:	7%
	Publications:	7%
	Student Papers:	5%

2% match (publications)
Ke, Shian-Ru, Hoang Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. "A Review on Video-Based Human Activity Recognition", Computers, 2013.

2% match (Internet from 14-Jan-2014)
http://people.cs.uct.ac.za

1% match (Internet from 26-Feb-2016)
http://epub.uni-regensburg.de

1% match (student papers from 02-May-2005)
Submitted to University of Sunderland on 2005-05-02

1% match (student papers from 14-Apr-2015)
Submitted to University of Abertay Dundee on 2015-04-14

1% match (publications)
Alper Yilmaz. "Object tracking", ACM Computing Surveys, 12/25/2006

<1% match (Internet from 08-Jun-2012)
http://www.orbitamerica.com

<1% match (publications)
Tomic, Katarina(Mayer, Christian and Veeman, Wiebren S.). "About drugs in hydrogel drug delivery systems", DuEPublico: University of Duisburg-Essen Publications Online, 2008.

<1% match (Internet from 08-May-2009)
http://www.lumes.lu.se

<1% match (Internet from 07-May-2003)
http://www-vlsi.stanford.edu

<1% match (Internet from 03-Mar-2015)
http://www.readbag.com

<1% match (Internet from 19-Jun-2014)
http://www.computing.edu.au

<1% match (publications)
Karaman, Mustafa. "Towards Robust Object Segmentation in Video Sequences and its Applications", Technische Universität Berlin, 2010.

<1% match (Internet from 19-Oct-2014)
http://luisteixeira.net

<1% match (Internet from 16-Dec-2015)
http://edtb.euskomedia.org

<1% match (Internet from 19-Feb-2015)
http://www.yongxu.org

<1% match (Internet from 21-Oct-2010)
http://www.food.lth.se

<1% match (Internet from 24-May-2012)
http://ieeexplore.ieee.org

<1% match (Internet from 10-Jan-2008)
http://bellota.ele.uva.es

<1% match (Internet from 18-Aug-2010)
http://www.soniacoleman.com

<1% match (publications)

<1% match (publications)
Na, Hana(Qin, S and Wright, D). "A study on detection of risk factors of a toddler's fall injuries using visual dynamic motion cues", Brunel University School of Engineering and Design PhD Theses, 2009.

<1% match (student papers from 18-Sep-2015)
Submitted to University of Wales, Bangor on 2015-09-18

<1% match (Internet from 11-Apr-2012)
http://www.isip40.it

<1% match (Internet from 13-Mar-2016)
http://etheses.bham.ac.uk

<1% match (publications)
Heide, Dominik. "Statistical physics of power flows on networks with a high share of fluctuating renewable generation", Publikationsserver der Goethe-Universität Frankfurt am Main, 2011.

<1% match (Internet from 24-Dec-2015)
https://publications.polymtl.ca/318/1/2010_ParisaDarvishZadehVarcheie.pdf

<1% match (student papers from 01-Sep-2014)
Submitted to Sheffield Hallam University on 2014-09-01

<1% match (Internet from 15-Jun-2009)
http://techpubs.sgi.com

<1% match (publications)
Bai, Shuang. "A Study of Informative Patch Extraction and Patch-level Context Exploration for Image Categorization", NAGOYA Repository (New oai_dc setup Aug 2014), 2013.

<1% match (publications)
Reckley, Adam R. Hsu, Wei-Wen Chen, Chun. "Sensor selection and integration to improve video segmentation in complex environments.(Research Art", Mathematical Problems in Engineering, Annual 2014 Issue

<1% match (publications)
Industrial Management & Data Systems, Volume 112, Issue 6 (2012-06-30)

<1% match (Internet from 04-Apr-2009)
http://www.cs.memphis.edu

<1% match (Internet from 22-Sep-2015)
http://eprints.qut.edu.au

<1% match (Internet from 17-Dec-2007)
http://www.mwra.com

<1% match (Internet from 01-Aug-2014)
http://www.eecs.yorku.ca

<1% match (Internet from 23-Sep-2010)
http://vision.stanford.edu

<1% match (publications)
International Journal of Web Information Systems, Volume 5, Issue 4 (2009-12-15)

<1% match (student papers from 05-Jan-2016)
Submitted to University of Bedfordshire on 2016-01-05

<1% match (Internet from 21-Feb-2015)
http://www.dri.edu

<1% match (Internet from 16-Sep-2012)
http://home.iitk.ac.in

<1% match (Internet from 23-Feb-2015)
http://www.docstoc.com

<1% match (Internet from 13-Dec-2015)
http://clok.uclan.ac.uk

<1% match (Internet from 03-Mar-2011)
http://www4.comp.polyu.edu.hk

<1% match (Internet from 09-Apr-2012)
http://howtofindslope.net

<1% match (publications)
Wan, Jun Ruan, Qiuqi An, Gaoyun Li, Wei . "The dynamic model embed in augmented graph cuts for robust hand tracking and segmentation in videos.", Mathematical Problems in Engineering, Annual 2014 Issue

<1% match (publications)
Industrial Robot: An International Journal, Volume 34, Issue 1 (2007-01-14)

<1% match (Internet from 30-Jan-2016)
http://ir.canterbury.ac.nz

<1% match (Internet from 08-May-2009)
http://www.utexas.edu

<1% match (Internet from 15-May-2015)
http://www.db-thueringen.de

<1% match (Internet from 15-Jun-2012)
http://kth.diva-portal.org

<1% match (Internet from 12-Oct-2010)
http://dilnxsrv.king.ac.uk

<1% match (Internet from 14-Aug-2012)
http://people.cs.uct.ac.za

<1% match ()
http://cvrr.ucsd.edu

<1% match (Internet from 20-Dec-2014)
http://faculty.ucmerced.edu

<1% match (Internet from 18-Aug-2011)
http://kedarpatwardhan.org

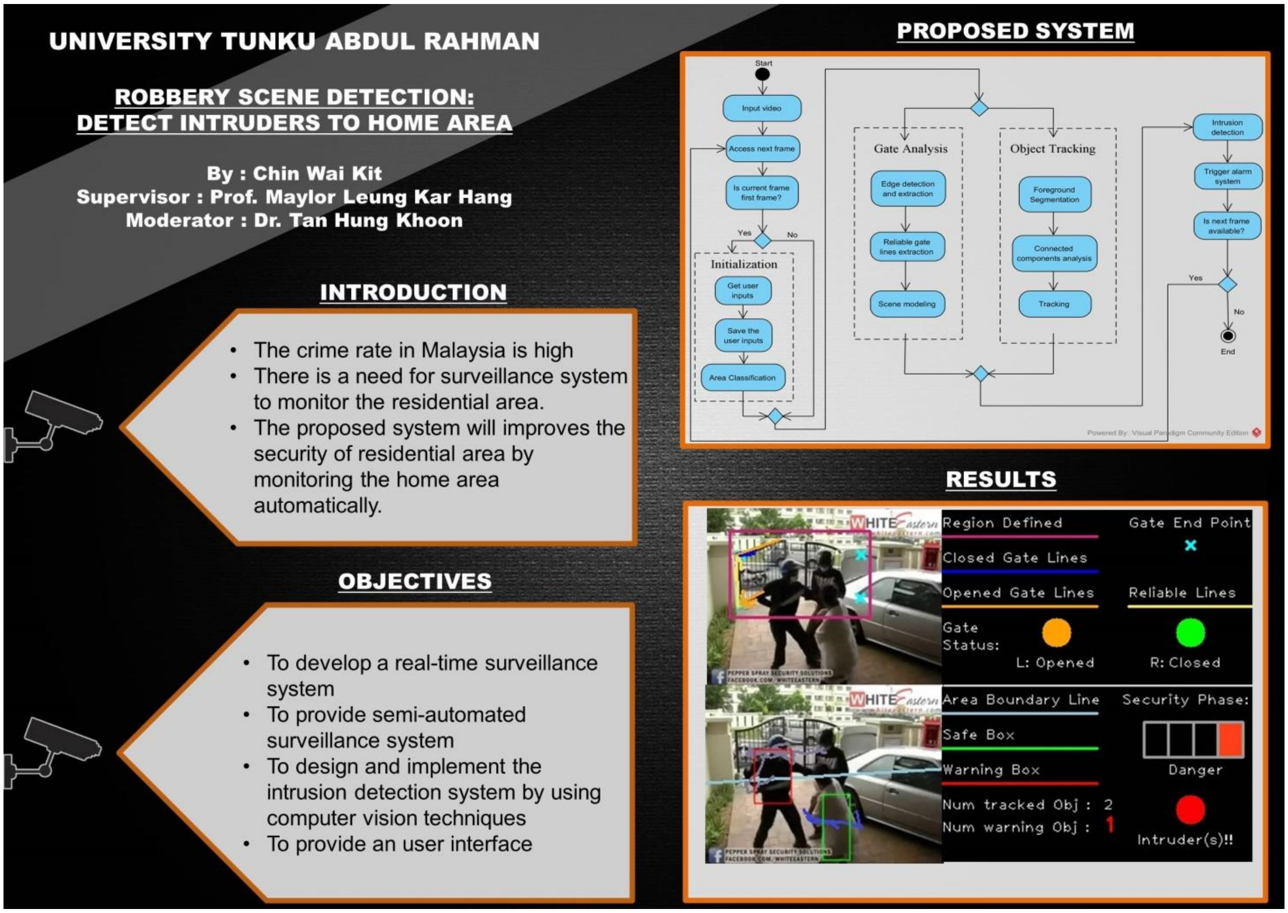
<1% match (publications)
João Manuel R. S. Tavares and Raquel R. Pinho. "Tracking features in image sequences with Kalman filtering, global optimization, mahalanobis distance and a management model", Repositório Aberto da Universidade do Porto, 2014.

<1% match (publications)
Liang, X, Moradi, F, Tan, G and Taylor, S J E. "An agent-based DDM for high level architecture", IEEE, 2009.

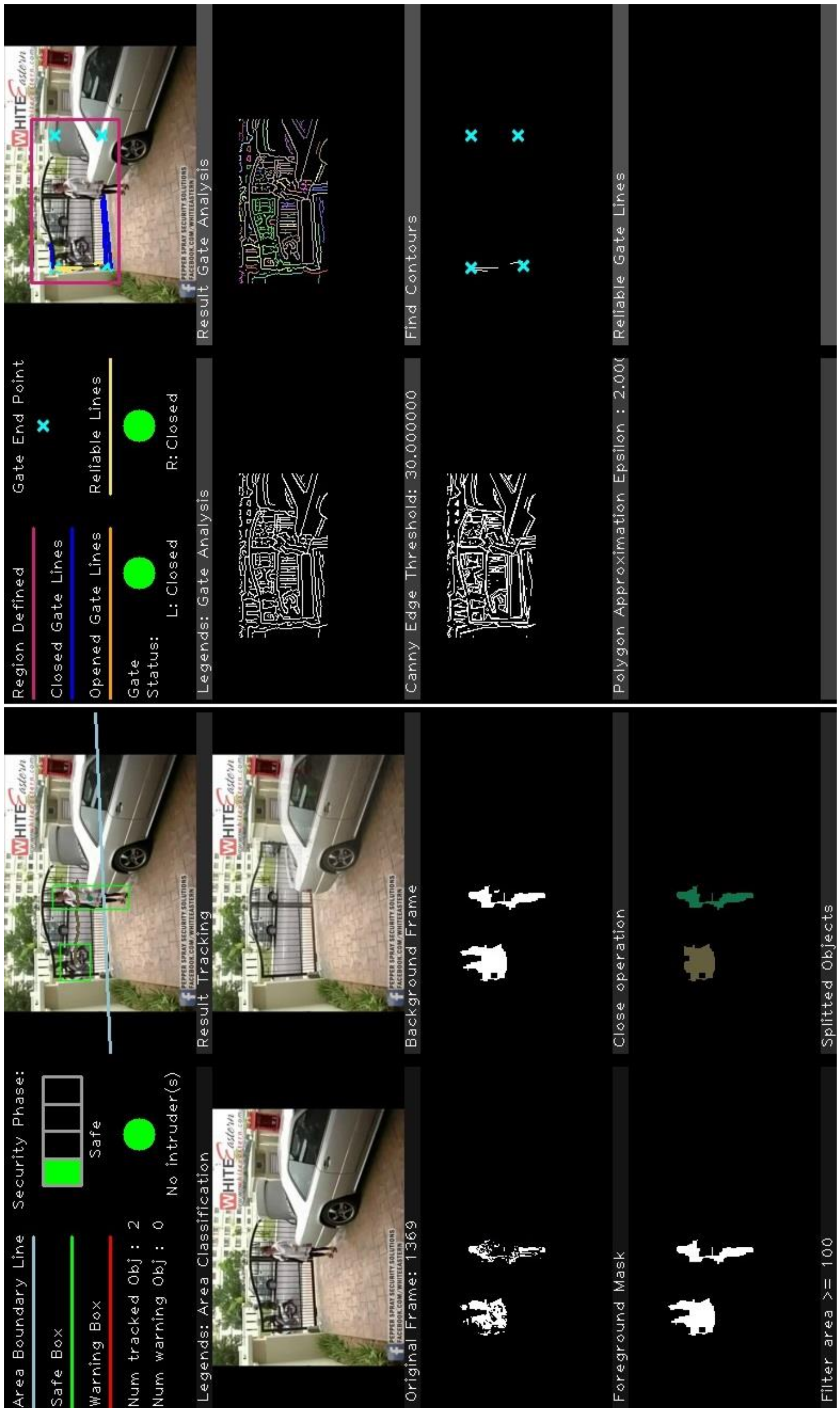
<1% match (publications)
Industrial Robot: An International Journal, Volume 40, Issue 6 (2013-10-19)

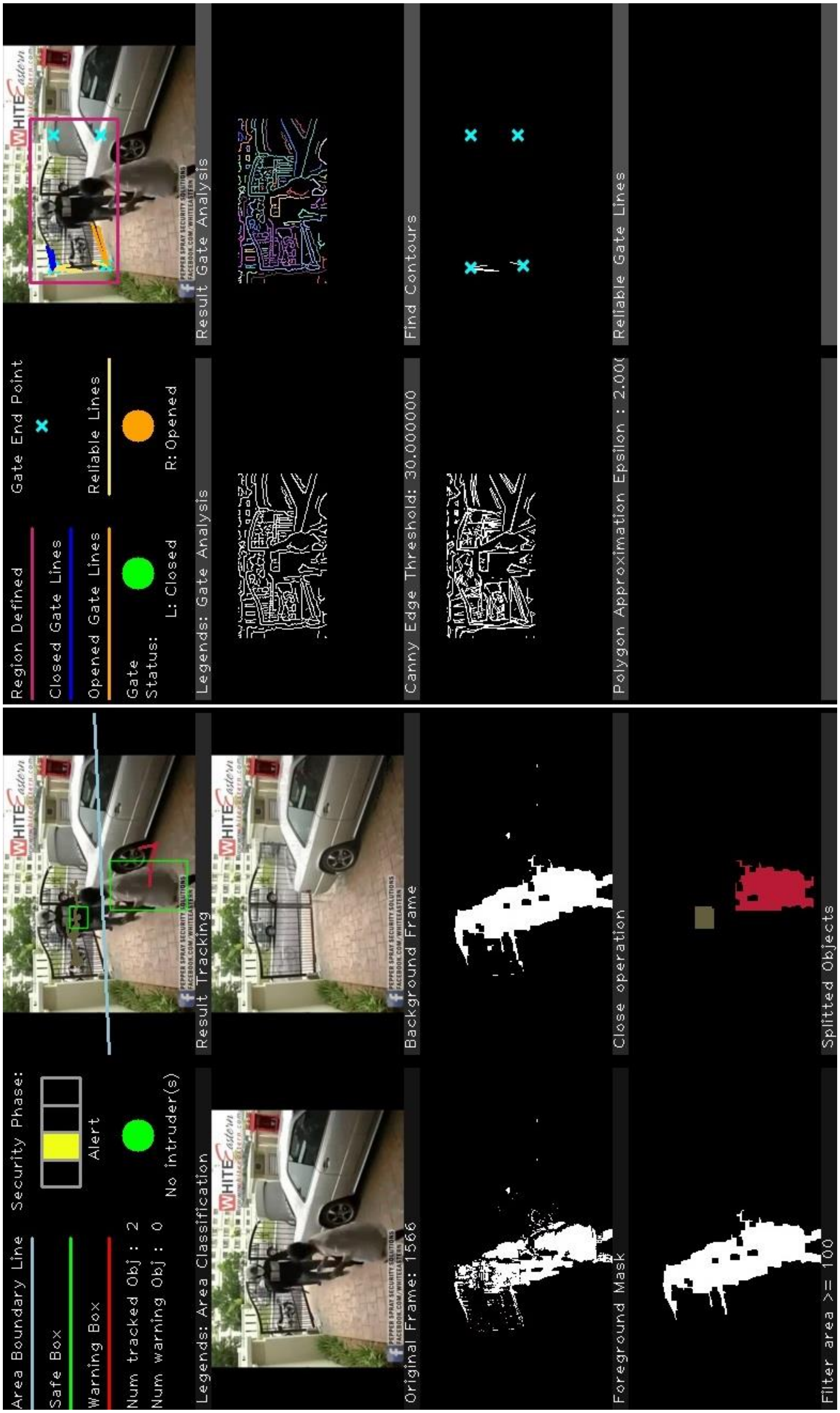
<1% match (publications)
Baskaran, Kumaran. "Chemical shift optimization and ensemble averaging in protein NMR spectroscopy", Publikationsserver der Universität Regensburg, 2011.

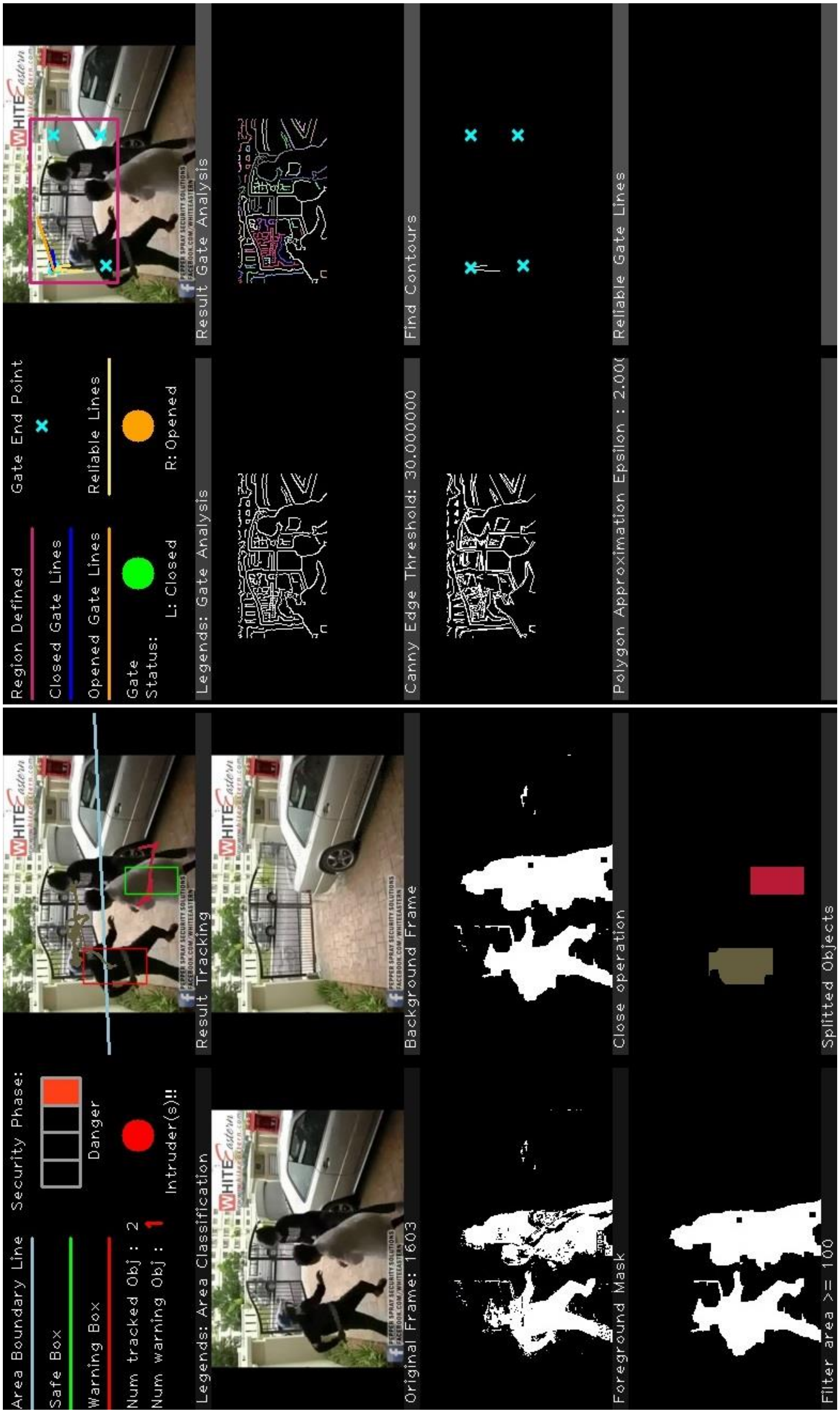
APPENDIX B POSTER

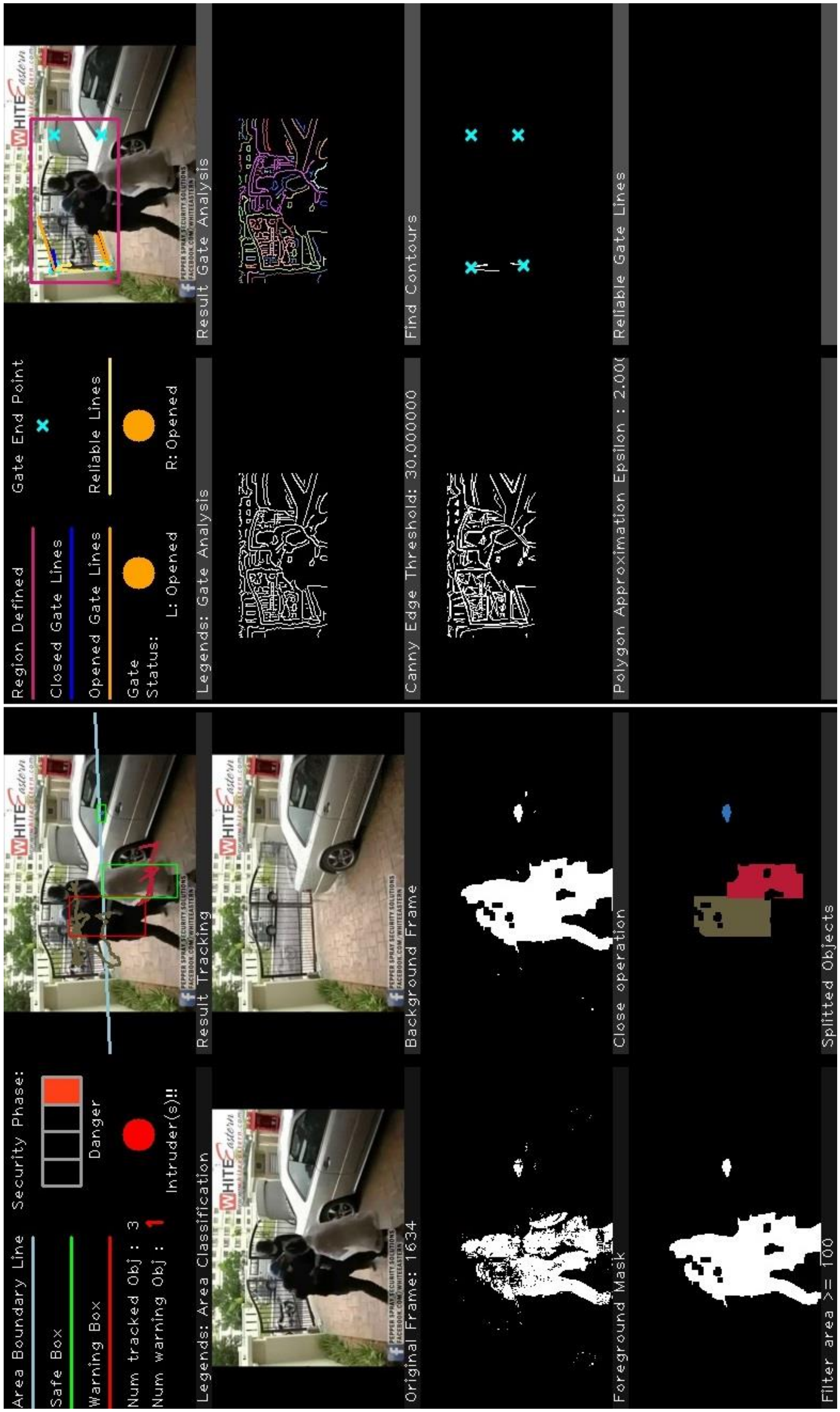


APPENDIX C RESULTS




Area Boundary Line
Safe Box
Warning Box
Num tracked Obj : 2
Num warning Obj : 0
Security Phase:
Alert
No intruder(s)
Legends: Area Classification
Result Tracking
Original Frame: 1566
Background Frame
Foreground Mask
Close operation
Filter area >= 100
Splitted Objects
Region Defined
Closed Gate Lines
Opened Gate Lines
Gate End Point
Reliable Lines
Gate Status:
L: Closed
R: Opened
Legends: Gate Analysis
Result Gate Analysis
Canny Edge Threshold: 30.000000
Find Contours
Polygon Approximation Epsilon : 2.00(
Reliable Gate Lines


Area Boundary Line
Safe Box
Warning Box
Num tracked Obj : 2
Num warning Obj : 1
Security Phase:
Danger
Intruder(s)!!
Legends: Area Classification
Result Tracking
Original Frame: 1603
Background Frame
Foreground Mask
Close operation
Filter area >= 100
Splitted Objects
Region Defined
Closed Gate Lines
Opened Gate Lines
Gate Status:
L: Closed
Gate End Point
Reliable Lines
R: Opened
Legends: Gate Analysis
Result Gate Analysis
Canny Edge Threshold: 30.000000
Find Contours
Polygon Approximation Epsilon : 2.00(
Reliable Gate Lines


Area Boundary Line
Safe Box
Warning Box
Num tracked Obj : 3
Num warning Obj : 1
Security Phase:
Danger
Intruder(s)!!
Legends: Area Classification
Result Tracking
Region Defined
Closed Gate Lines
Opened Gate Lines
Gate Status:
L: Opened
Gate End Point
Reliable Lines
R: Opened
Legends: Gate Analysis
Result Gate Analysis
Original Frame: 1634
Background Frame
Canny Edge Threshold: 30.000000
Find Contours
Foreground Mask
Close operation
Polygon Approximation Epsilon : 2.00(
Reliable Gate Lines
Filter area >= 100
Splitted Objects