SECURITY ALERT THROUGH PATH ANALYSIS

By

KELVIN LO YIR SIANG

A dissertation submitted to the Department of IPSR,
Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Computer Science
November 2013

ABSTRACT

SECURITY ALERT THROUGH PATH ANALYSIS

Kelvin Lo Yir Siang

The purpose of this project is to develop a computer vision system that is capable of clustering the path that is taken by the human that is detected in the scene more efficiently in order to create a path model for analyzing purposes. The proposed system's internal function will be triggered if there are any moving objects in the scene and will be using path clustering technique for clustering similar path and cluster index searching technique for searching to correct path to cluster with. The moving objects will be classified as human or non-human. Non-human moving object will be ignored but human moving object will be keep track of until that human moving object leaves the scene.

The proposed system consist of 3 modules; Human Detection Module, Path Detection and Clustering Module and Suspicious Path Detection Module. The first module is responsible for classifying the moving object into human and non-human group. This is done by using Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM). The detected human will then be tracked by using Optical Flow. The second module will be responsible for detecting and clustering similar path. This module will only run during the training stage. A combination of path clustering technique and a cluster index searching technique will be used. The third module is responsible for analyzing the path that is taken by the detected and tracked human. This module will only run during the testing stage. If the path that is taken by the tracked human in the scene is not commonly used by others, the proposed system will alert the security personal.

ACKNOWLEDGEMENT

Looking back, I am surprised and at the same time very grateful for all I have received throughout these years. It has certainly shaped me as a person and has led me where I am now. All these years of Master studies are full of such gifts.

I want first to thank my supervisor Dr. Khor Siak Wang for all the hope he has put on me, before I thought I could do any research at all. He has always encouraged me to live intensively, even when I was in a slump and can’t think of doing anything else. During more than three years of knowing him, he has helped me with both my education and my research. He also taught me how to appreciate the good scientific work and has enlightened me through his wide knowledge of Artificial intelligence and his deep intuitions about where it should go and what is necessary to get there.

I also would like to thank Dr. Tay Yong Haur, my co-supervisor for all the help and knowledge sharing that he has given to me generously. Though he has only been my co-supervisor for a little more than a year, his knowledge that he shared with me through various discussions that we had has been proven to be quite vital to my research. He has also been quite a friendly, helpful and approachable co-supervisor. He will answer all the uncertainties that I have in my mind and clear any doubts that I have.

Next, I would like to thank my beloved family. My father, John Lo Vun

Nyen, has always given me moral supports and encouragement of any form during my Master studies. He would always be there in my speed dial and will always answer my call for help no matter how busy he may have been. He would always reserve a time for me even in his busy schedule. My mother, Teresa Chang Li Yun, has always given me advice, love and support when I was at my weakest state in both mind and body. I have lost count the number of times I have rebound back to my best state from my worst state through my mother who have been continuously ensuring me and giving me confidence. My sister, Stephanie Lo Yir Tiane, has always been there for me whenever I need her. She is the one who have always been there to pick me up when I fall. She too has been giving me advice and encouragement during my Master studies. Best of all, she was the one whom I always unleash my frustration and stress upon. I would like to thank her for listening to all my complaints patiently.

For the period of time I spend during my Master studies, I am blessed with friends who share similar interest in both hobbies and interests. When I am extremely down and was stuck in a slump, they would try their best in their own way to cheer me up and lift my spirit. Therefore, I would like to thank my group of friends and the people in the same computer lab for helping me to stay centered and rediscover what is important and why I am doing what I am doing.

Next, I would like to give a special thanks to my ex-supervisor Dr. Lee Yun Li and Dr. Wang Xin who has been helping me out with various problems that I faced during my Master studies. Last but not least, I would also like to specially mention my thanks to MIMOS, who have provided me with the means and the tools to do my research.

APPROVAL SHEET

This dissertation/thesis entitled "**SECURITY ALERT THROUGH PATH ANALYSIS**" was prepared by KELVIN LO YIR SIANG and submitted as partial fulfillment of the requirements for the degree of Master of Computer Science at Universiti Tunku Abdul Rahman.

Approved by:

____
(Dr. Khor Siak Wang)   Date: …………………..
Supervisor
Department of Information Systems
Faculty of Information Communication and Technology
Universiti Tunku Abdul Rahman

____
(Dr. Tay Yong Haur)   Date:…………………..
Co-supervisor
Department of Internet Engineering and Computer Science
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___18 NOVERMBER 2013____

SUBMISSION OF THESIS / DISSERTATION

It is hereby certified that **KELVIN LO YIR SIANG** (ID No: **10UEM07395**) has completed this thesis/dissertation entitled "SECURITY ALERT THROUGH PATH ANALYSIS" under the supervision of DR. KHOR SIAK WANG (Supervisor) from the Department of Information Systems, Faculty of Information Communication and Technology, and DR. TAY YONG HAUR (Co-Supervisor) from the Department of Internet Engineering and Computer Science, Faculty of Engineering and Science.

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

Yours Truly,

____

(KELVIN LO YIR SIANG)

DECLARATION

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

Name ____KELVIN LO YIR SIANG____

Date ______18 NOVERMBER 2013_____

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CCTV	Closed Circuit Television
CA	Clustering Accuracy
CCR	Correct Cluster Rate
DTW	Dynamic Time Warping
ELD	Enhanced Levenshtein Distance
GMM	Gaussian Mixture Model
GEMINI	Generic multimedia object indexing
HOG	Histogram of Oriented Gradient
HOT	Histogram of Template
LSVM	Linear-SWM
LAN	Local Area Network
LCSS	Longest Common Subsequence
MoG	Mixture of Gaussian
PCA	Principle Components Analysis
PDF	Probability Density Function
ROI	Region of Interest
SVM	Support Vector Machine
TP	Tracking Point

CHAPTER 1

INTRODUCTION

1.1 Research Background

Anything can happen anywhere at any time. Accidents or crime could occur suddenly and the ability of security personnel to react upon such incidents is crucial. A very good example would be approaching an individual to enquire the reason for loitering around within an area where no one should enter or maybe leaving an area quickly looking worried as soon as such incident is detected by the surveillance system. (C. Stauffer and E. Grimson, 2000) Individuals who are acting suspiciously usually has a motive and security personnel should be more vary of such individuals and able to react quickly.

This, combined with the inadequate of security control over crime, has prompted outrage among its citizens enough to take justice into their own hands in apprehending criminals. (Arnold Wiliem et al., 2009) Thus a research is being conducted to solve this problem. Tracking human is of primary interest for many applications such as surveillance, action monitoring, and path detection. (I.Junejo et al., 2004) The goal is to learn the routes or paths most commonly taken by humans as they traverse through a scene and register any unusual activity. An example of an unusual behavior might be a person walking in a region not used by most people or a person following a zigzag path. A path is any established line of travel or access. Most human follow a common trajectory while entering or exiting a scene due to presence of pavements, benches, or designated pathways. Therefore, a method is required that can model the usual trajectories of the human and indicate a typical trajectories that might call for further investigation through any higher level event recognition.

This research is concern primarily with pedestrian movements across a scene but the method is general and can be extended to any scenario. The motivation for such a system is manifold. Primary application of this method is video surveillance. At many public places, like an airport, the people must be made sure to be kept away from a certain area or on a street where one might want to make sure no one is walking drunk. Moreover, as common pathways are detected by clustering the trajectories of human, detected trajectory can be efficiently assign to its associated generic topographical scene description known as “path model”.

Therefore, a system that could group similar trajectories together is much desired. This research project is aimed to develop a system prototype which is capable of detecting human in the targeted scene, cluster the trajectories of human and efficiently assign it to the associated generic topographical scene description. As this proposed system can be implemented in the society, it can help to prevent and reduce the crime rate in our nation.

1.2 Problem Statements

1.2.1. Path clustering is computational intensive

Grouping all the similar paths, i.e. path clustering, into clusters is an essential step to guide the detection of unusual paths. Common techniques for such clustering include similarity based clustering, statistically based model of object trajectories clustering, Longest Common Subsequence (LCSS) distance, Hausdorff Distance, Hidden Markov Model-Based Distance and Euclidean Distance Method (Zhang Zhang et al., 2006). The most commonly used and most accurate method used for path clustering is the Euclidean distance Method that is implicated in Levenshtein Distance method. The Levenshtein Distance algorithm that uses Euclidean Distance repetitively is very computational intensive (E.J.Keogh and M.J.Pazzanin, 2000).

1.3 Objectives of the Research

1.3.1. Improve the efficiency for path clustering

Such a reduction can be done by only considering specific points rather than all the points that lie within the path. With such reduction, it is able to speed up the entire system performance. Furthermore, by implementing the new searching method for similar cluster in the generic topographical scene description, the similar cluster can be found more efficiently. By using the old method of searching, it is required to compare with the entire available cluster in the generic topographical scene description in order to search for the similar cluster. However, if the new searching method is implemented, certain cluster that does not meet the condition can be ignored and thus the searching can be done more efficiently.

1.4 Project Scope

The main purpose of this research project is to automatically raise a security alert whenever there is any unusual path taken by pedestrian within a monitored scene. In order to achieve this, the entire research project is broken down into three major modules as follows:

1) Human Detection
2) Path Detection and Clustering
3) Suspicious Path Detection

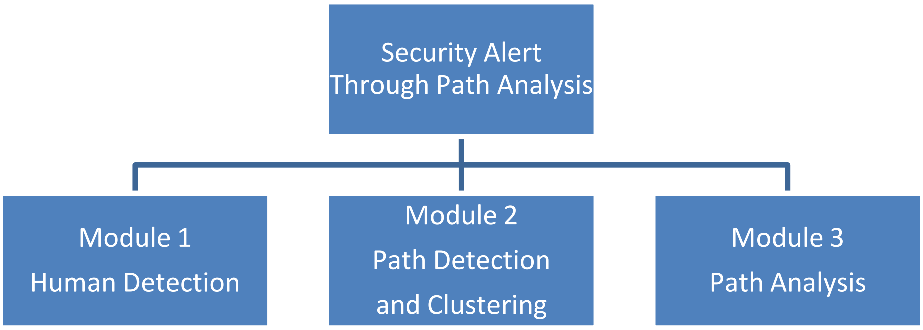

Security Alert Through Path Analysis
Module 1 Human Detection
Module 2 Path Detection and Clustering
Module 3 Path Analysis

Figure 1.1: System Module Break Down

1.4.1. Module 1 – Human Detection

The first Module is for detecting human that appears in the video. There could be many other motions in the video captured by the camera but not all these motions are humans. Thus, this module will be used to filter out these motions and determine which of them are humans are and which are not. The motion that is labeled as human will then be tracked to be used in the second module described below.

1.4.2. Module 2 – Path Detection and Clustering

Generally, this module is for training purpose where a generic topographical scene description will eventually be produced based on all the identified clusters. Here, al the paths taken b pedestrians will be captured, extracted and placed into correct clusters. All these clusters will form a generic topographical scene description that will be used as a base for reference, i.e. help in locating those unusual paths that could be possible is taken by unusual pedestrian into a restricted area.

1.4.3. Module 3 – Suspicious Path Detection

This is where the actual detection of unusual path is performed. The generic topographical scene description or also known as “Path Model” produced in module 2 will be used to guide such detection. Whenever a fresh path is detected, a constant check with those clusters in the generic topographical scene description will be performed. Security alert will be triggered should this fresh path not fall within those clusters.

1.5 Dissertation Organization

This dissertation is organized into 6 chapters.

Chapter 1 will be the introduction to the research project. Project background, problem statement, objectives, project scope, research methodology, potential application, contributions and also the Gantt chart of this project will be presented.

Chapter 2 will focus on the literature review. Relevant literatures will be studied and will be presented in this chapter. A brief discussion and comparison of literatures will also be performed.

Chapter 3 is about the system design, methodology of the proposed system and also the system implementation which includes part of the system code, screen shot of the system interface, and also the discussion of testing environment. This chapter will review about how the system's process flow, the way the system works and also the characteristics of the system.

Chapter 4 is the about the discussion of the contribution of this research. A detail explanation of the contribution will be shown in this chapter.

Chapter 5 will discuss the result of the proposed system. The result of performance evaluation of the system will be shown and discussed in this chapter.

Chapter 6 will discuss the future work and conclude this research project. A summary of the whole research project will also be included.

1.6 Research Methodology

The first stage will be the literature review. In this stage, works and papers that are related to the research project written by other scholars will be reviewed. The second stage will be the data collection stage where all related data such as training and testing data are collected. The third stage is the development stage. This is where the prototype of the system will be developed. The fourth stage is the experimental stage. This is the stage where the prototype that is developed in the previous stage undergoes a series of testing using the data collected in previous stage.

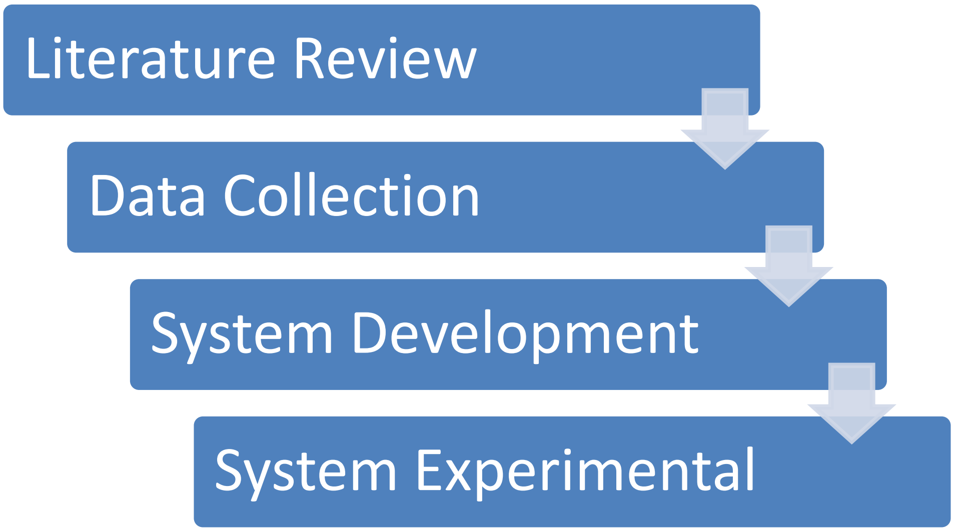


Figure 1.2: Research Methodology

1.7 Proposed Work Schedule

Activities	Period	June 2010 to May 2012																		
	(month)	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
Topic research and proposal writing	3	■	■	■																
• Literature reviews		■	■																	
• Writing proposal			■	■																
Analysis and development of prototype system	14				■	■	■	■	■	■	■	■	■	■	■					
• System planning and analysis					■	■	■													
• Data collection								■	■	■	■	■	■							
• System design and development											■	■	■	■	■					
• System testing and improvement														■	■					
Finalizing the system and report writing	7															■	■	■	■	■
• Compilation and evaluation of whole system																■	■	■		
• Write complete report and papers																			■	■

Figure 1.3: Gantt Chart

1.8 Expected Research Outcome

The expected outcome for this research is a surveillance system prototype that is capable of detecting human in the video and implementing a new and improved technique for path detection and clustering in order to improve the system’s performance. Through the literature review, the methods that are implemented by other scholars will be used and implemented if their methods are more efficient.

1.9 Contribution

For this project, the contribution will be the new and improved method for clustering similar path and a more efficient method of searching for similar path cluster in the generic topographical scene description.

1.10 Potential Application

The surveillance system can be applied in shopping malls where the number of scenes that need to be monitored is enormous and has only a limited number of guards on duty to monitor the scenes. Criminal activity could be prevented and also a quick and accurate action can be taken in a different kind of situation such as coordinating the security personals in catching the criminal in the shopping mall.

1.11 Conclusion

This system’s aim is crime prevention and reduction by alerting the security about suspicious individuals that are shown on the monitor. It is also a security system that is aimed to protect the society from being the victims of the crime. With that being said, with the creation of this system, it is hoped that the crime rate will drop and will allow more peaceful and harmony living environment. As a conclusion, such research is not only important but is also worthy of our time.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

By taking the vast number of postures and movement that can be done by human into consideration, the number of actions that can be taken by a human is enormous. This undoubtedly poses a great challenge for a system to detect human. A security system without the ability to detect human or the ability to properly recognize any human activity will be useless as any moving object will be considered as a security risk. Thus, to solve such problem, countless of researchers have put in all their efforts to try and come out with an actual working human detection module. As stated in the previous chapter, this research comprises of 3 major modules; Human Detection module, Path Detection and Clustering module and Suspicious Path Detection module. Thus, all relevant research that is done will be reviewed and a comparison of their methods will be performed in this chapter.

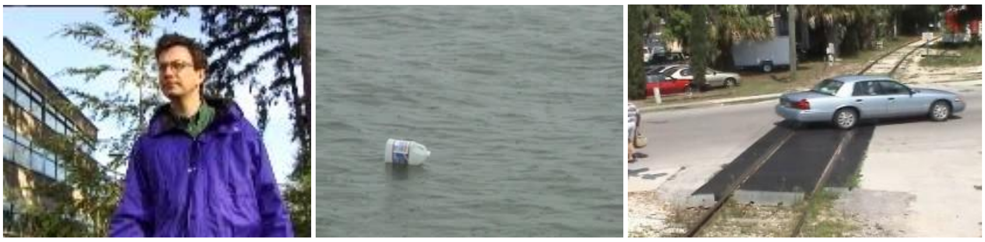
2.2. Human Detection Module

For this module, it can be divided into three major sections, namely Background subtraction, Human Detection, and Human Tracking. Each sections the literature review on particular field of research, as follows:-

- Background subtraction
- Human detection
- Human tracking

2.2.1. Background Subtraction

Foreground segmentation, or commonly known as background subtraction, is a fundamental technique used for video content analysis with the purpose of extracting foreground objects e.g. moving objects from the captured scenes. This procedure provides necessary low-level visual cues to facilitate further analysis, such as object tracking (Kim, 2008), action or activity recognition, and is crucial to many applications including surveillance systems, human computer interaction, animation and video event analysis (Cheng and Gong, 2009). By subtracting the background image from the current video frame, the motion (foreground) object can be detected easily. A number of researches towards background subtraction technique have been done over the years. many challenges on this field were addressed. The most common one would be the problem where the video background scene is dynamic. A dynamic scene can be defined as a scene that contains significant background motions (Chan et al., 2010); natural scenes compose of several dynamic entities; or complicated backgrounds that are themselves moving, e.g. swaying trees, moving water, waves and rain (Mahadevan and Vasconcelos, 2008). Figure 2.1 shows some sample of dynamic scenes where (a) is Wallflower dataset (Toyama et al., 1999); (b) is jug dataset (Zhong and Sclarroff, 2003); and (c) is railway dataset used in Sheikh and Shah's research (2005).



(a) (b) (c)

Figure 2.1: Sample dynamic scene

Common techniques/approaches used to perform foreground segmentation include:

- Texture-based Method
- Hypothesis-based Method
- Background Modeling-based on Gaussian Mixture Model (GMM)
- Codebook-based Background Subtraction
- Frame Differencing Approach

2.2.1.1. Texture-based Method

Heikkilä and Pietikäinen (2006) presented a novel and efficient texture-based method for modeling the background and detecting moving objects from a video sequence which made use of the discriminative texture features to capture background statistic. Unlike most other approaches, the features in background modeling are computed over a larger area than a single pixel which could offer with many advantages and improvements compared to the state-of-the-art. However, one of the major drawbacks of the proposed method is that it could not perform handling of shadows which had turned out to be an extremely difficult problem to solve with background modeling. The experiment results had shown that the proposed method performed better in terms of lesser false negatives and false positives, as compared to the Gaussian Mixture Model (Stauffer and Grimson, 1999).

2.2.1.2. Hypothesis-based Method

Mahadevan and Vasconcelos (2008) proposed a new paradigm for background subtraction in dynamic scenes. However, the authors equated the background subtraction to the problem of detecting salient motion, and proposed a solution based on a generic hypothesis for biological salience, which is referred to as the discriminant center-surround hypothesis. Under such hypothesis, bottom-up saliency was formulated as the result of optimal discrimination between centers and surround stimuli at each location of the visual field. Locations where the discrimination between the two could be performed with smallest expected probability of error were declared as most salient. Background subtraction was then equivalent to simply ignoring the locations declared as non-salient. Furthermore, this research work was aimed to deal with extremely dynamic backgrounds, thus in the experiment the method outperformed all the other techniques which mostly only dealt with background with minimal dynamics.

McHugh et al. (2009) revisited background subtraction from the hypothesis testing point of view, and made two contributions. Assuming spatial ergodicity, the authors augmented the background model with an explicit foreground model and estimated its parameters from a small spatial neighborhood which was unlike previous approaches where the background model's probability density function (PDF) was assumed uniform or was estimated from several past frames. The new model improved the method's discrimination sensitivity at the cost of a slight increase in computational complexity. Also, the authors embedded a Markov model into the hypothesis test to spatially characterize the change of labels. The proposed approach was using a spatially-variable detection threshold that offered an improved spatial coherence of the detections.

2.2.1.3. Background Modeling-based on Gaussian Mixture Model (GMM)

Zhong et al. (2009) also proposed a multi-resolution background subtraction method to deal with dynamic scenes. Gaussian Mixture Model

(GMM) was incorporated. Making use of the advantages in characteristics in both original resolution and low resolution image, the proposed approach was able to extract foreground object with low false positive and false negative results. Wang et al. (2009) introduced an approach to build the background model based on the common Mixture of Gaussian (MoG). They made use of the unique characteristic of pixel in the same image segment to re-formulate the general background model and combined with a simple and fast fusing mechanism to obtain a smoothed Mahalanobis distance map and finally obtaining the foreground masks by constructing and minimizing an energy function. The method was tested with five previously used video sequences and the results had proven the consistent performance of the method.

2.2.1.4. Codebook-based Background Subtraction

Pal et al. (2010) modeled the background using the codebook algorithm first proposed by Kim et al. (2005) which was based on a clustering stage. The approach was tested extensively and the results had shown that it outperformed the original codebook-based algorithm. Meanwhile, the authors discussed the problem caused by dynamic background such as swaying trees, which was always a big challenge in background subtraction.

2.2.1.5. Frame Differencing Approach

In the works by Xia et al. (2010), frames differencing method was used. The authors had proposed an approach which combined background modeling with frames differencing method for the purpose of detecting foreground objects. The background model was updated using average method which depended on the means of the current image and last background image. An assumption that noises in the image agreed with the Gaussian distribution was made. Due to many limitations in average method, the authors made use of frames differencing method. Performing frame-to-frame subtraction over a sequence of frames and performed OR operation for the resulting image containing sequence of moving object’s contour. The final results were then obtained by performing AND operation to both the background subtraction and frames differencing result.

Lian et al. (2010) proposed a novel method on moving object detection based on background subtraction and three frames differencing. In the image acquisition and transmission process, an image pre-processing process was performed which applied spatial median filter to smoothen the input image data. In the algorithm, the authors constructed the background model through learning the new available frames and updated the parameters of Gaussian model corresponding to each pixel by taking into account the past background information. Besides, by using three frames temporal differencing method, moving object could be obtained after searching for the overlapping part in the grayscale image. The experiment results had shown that the proposed algorithm not only worked well in real time processing, but also able to extract moving-objects in the image effectively. This proposed method was effective in overcoming the poor adaptability of background subtraction, however, it is poor in the background update when gray level of moving-objects and the background were close or in case of sudden change of illumination.

2.2.2. Human Detection

Human Detection or Pedestrian Detection is a rapidly evolving area in computer vision with key applications in intelligent vehicles, surveillance, and advanced robotics (Enzweiler and Gavrila, 2009). The main purpose of human detection is to classify whether an object is human or non-human objects. In the context of a surveillance system, the reason of making such classification is that human is the only object of interest which could possibly cause security threats.

Common way of performing human detection is to use global features of human body to detect pedestrians. Besides, alternate features are also proposed by different researchers to improve/enhance the human detection results. In this literature review, features used to detect human are as follows:

- Global Feature-based
- Local Feature-based

2.2.2.1. Global Featured-based

Support Vector Machine (SVM) (Vladimir, 1995) is widely used in human classification due to its advantage where it is suitable to compare the discriminative ability of different features (Tang and Goto, 2009). Cao et al. (2008) proposed a low-cost pedestrian detection system using SVM classifier. Besides, AdaBoost algorithm was also used in the training procedure. The system adopted cascaded classification using both the statistical learning and the decomposed SVM classification. The system detected pedestrians using cascade classifier that used both appearance and motion features. The experiments had shown that the proposed algorithm had a very good performance in both the pedestrian detection and the direction identification.

Ding et al. (2009) proposed an algorithm for unsupervised spatio-temporal multi-human detection and recognition in complex scene, executed from spatial domain and time domain. The proposed algorithm consisted of four main detection modules which were target detection, target size estimation, human classification using SVM within current frame, and data association judgment among frames. The combination of spatial domain and time domain was a decision-making mechanism that unified target size estimation, SVM classification and data association judgment. The results had shown that the proposed algorithm presented good robustness and high accuracy, as well as real time capability.

In recent researches on human detection field, Lee et al. (2010) proposed an approach for human detection using Histogram of Oriented Gradient (HOG) descriptors. HOG feature set was inspired by Dalal and Triggs (2005), which had been widely used as human descriptor. The proposed algorithm by Lee et al. (2010) made use of SVM as the training and detection classifier. Besides, a human ratio estimation approach was used to filter unwanted objects such as noises, vehicles, etc which did not possess the same ratio as human. In the work by Choo et al. (2010), similar method was also used to detect human based on HOG feature and classification by SVM. Positive train data containing human and negative data containing non-human objects were put into the training procedure to train the classifier for human detection. Moreover, by applying a virtual mask to the scene, allowing the system to avoid waste of computation on uninteresting region, e.g. sky, on the wall, etc. However, such virtual mask was also a disadvantage for the approach where different masks were required for different scenes.

Jorge Baranda et al. (2008) proposed to use an algorithm that aims to detect humans by detecting human upper body to reduce the occlusion in indoor environment. They decided to use a linear-SWM (LSVM) as a classifier by which each dimension of the training and testing vectors was scaled to be between 0 and 1. Furthermore, in order to improve the performance of the system, they decided to reduce the number of HOG feature vectors for each object representation. This is being done by removing parts of the feature vectors that possessed less discriminative power. After some testing, result shows that the proposed method can reduce the feature length by half and the performance is improved in terms of speed and memory usage.

2.2.2.2. Local Featured-based

Instead of using global features, Tang and Goto (2009) proposed a human detection method based on motion and appearance based feature because they claimed that motion caused by human was different from motion caused by other objects. Optical flow was calculated to represent motion and feature was extracted from optical flow field and appearance based feature was extracted from standing human detection. Lastly, SVM was used for the result classification.

Xu et al. (2007) proposed a method which used some key local features of human body to help in pedestrian detection. To ensure the introduction of additional features do not slow down the detection speed, the authors used both appearance and motion global features of human body to select candidates and after that perform verification using local features of head and legs.

There is a new human detection method that is proposed by Csaba Beleznai Horst Bischof that is said to be faster than the average human detection method. (Csaba Beleznai Horst Bischof, 2009) This method uses the shape-based matching and motion cues. The said method is divided into 2 parts. The first part uses the template-based matching technique on the preprocessed contour image to locate the possible human while the second part is the local shape descriptor which inferred the human locations in images of absolute difference between current image and background. By combining the output from these 2 parts, it is possible to locate and detect the human.

Nguyen Duc Thanh, Philip Ogunbona and Wanqing Li proposed a new template matching approach that included matching and verification stages. (Nguyen Duc Thanh et al., 2009) Firstly, a set of predefined template images that described human postures will be used to find the closest matching description for the image that is extracted in the detection window. By definition of the authors, the closest matching template will be the one that have the shortest distance to the image within the detection window. With this output for the matching stage, the template will then be passed on to the next stage, verification stage where the template will then be checked to see whether it reached high degree of confidence that the image contains human. If, it returns positive, that the image has high credibility and high matching score, then the image will be classified as a human.

There is a part-based approach for human detection that was proposed by Shaopeng Tang and Satoshi Goto who used HOG features and boosting SVM classifier. (Shaopeng Tang and Satoshi Goto, 2009) Each classifier worked only to detect specific human parts such as head and shoulder, torso, leg, left of body and right of body. The cascade of classifier was designed so that for an object to be detected as human, it must pass all the cascades. In addition, area information was used for measuring the relative depth of humans to detect the human’s location for occlusion detection. In this way, even some part of the human was not detected, it still could pass the classifier and detected as occluded human.

In another paper, Shaopeng Tang and Satoshi Goto proposed the used of 2 different methods in detecting human. (Shaopeng Tang and Satoshi Goto, 2009) These humans could either be standing or moving. In order to detect standing human, they used the appearance based feature, HOT feature and SVM classifier while in order to detect moving human, motion based feature is extracted from the optical flow field that was calculated from the two consecutive frames (previous frame and current frame). The motion-based feature was about the relative motion of human body. It focused on local motion of human such as motion of legs and arms. After several testing, the result showed by combining both features, it is possible to achieved higher detection rate when compared to other features. Furthermore, this feature was not only hardware friendly but also possible to boost the detection rate by increasing the acceleration of the hardware.

In this paper Histogram of Template (HOT), a more efficient and discriminative feature was proposed by Shaopeng Tang and Satoshi Goto. (Shaopeng Tang and Satoshi Goto, 2010) The HOT was used together with a SVM classifier for human detection. Instead of using gradients, they used pre-defined templates for feature calculation and all templates were in the form of 3×3 grid. There were 3 pixels in each template. Should the value of the pixels satisfy certain predefined function, it means that the central pixel meets that particular template. Finally, the count of pixels which met a template would be collected to construct a HOT for human detection. After testing, the experimental result showed that the HOT feature is more discriminative yet having smaller feature size than HOG feature.

Lijing Chang and Yingli Liang proposed a new method to detect moving objects based on background subtraction method and used the shape analysis method for human detection. (Lijing Chang and Yingli Liang, 2010) According to them, there exists a lot of powerful classifier and feature representation in human detection. However, pre-processing the image is far more important and useful in most image processing approach. With that said their method of shape analysis made use of the features of motion region for detection. If the object area was larger than the pre-defined threshold and the aspect ratio of the object region satisfied the pre-defined ratio, then the moving object would be considered as a moving human. After testing, the result shows that this approach is not only faster and simpler; it is also able to remove shadows.

2.2.3. Human Tracking

According to Bhuvaneswari and Rauf (2009), object detection and tracking was a fundamental problem of computer vision research and very important for many real-life applications such as visual surveillance and human computer interaction. In particular, object tracking in video sequences was challenging under uncontrolled conditions (Wang and Yagi, 2008). The tracking algorithms would have to estimate the states of the targets when variations of background and foreground existed, occlusions happened, or appearance contrast became low. Trackers had to be efficient and could track variant targets. In this section, literature review will be done based on the problem of performing human tracking in computer vision system.

Common ways of performing object tracking are as follows:

- Feature Matching and Grouping
- Mean-Shift Algorithm
- Covariance Matrices Matching
- Kalman Filter
- Optical Flow Tracking

2.2.3.1. Feature Matching and Grouping

Javed and Shah (2002) developed and presented a method to track and classify objects in realistic scenarios. Object tracking in a single camera was performed by using background subtraction, followed by region correspondence. For each frame, moving objects were detected and defined as “region”. Regions, as compared to points, carried extra information like shapes and size. Each region was defined by the 2D coordinates of the centroid, X, the bounding box, B, and the size, S. The region for which correspondence had been established, had an associated velocity, V, and predicted change in size, ∇S. The tracking was done by searching for correspondence for all regions existed in frame T and $T-1$ using a cost function. The cost function took consideration into the differences of centroid location, as well as size of the region. The authors dealt with problem of uninteresting objects showing up as foreground region by using a minimum initial observation parameter O_{min}. If an object disappeared in less than O_{min} frames then it was considered as false detection. Similarly, a maximum missed observation parameter M_{max} was introduced to capture situation where objects temporarily disappeared from the scene. Occlusion was dealt by predicting new centroid location based on the velocity vector V.

Kanhere et al. (2005) introduced an approach which combined background subtraction and feature tracking. The author used the background subtraction result to estimate the 3D heights of corner features for a low-angle camera by assuming that the bottom of the background subtraction region was the bottom of the object. However, this kind of assumption often failed when occlusion occurred. Similarly, Kim (2008) proposed a similar approach which combined the background subtraction and the feature tracking and grouping algorithms. The authors used an augmentation to the background subtraction algorithm that used a corner feature tracking as a cue, and the feature tracking and grouping algorithm that used the result from background subtraction as a cue. Promising results were presented with various transportation applications.

2.2.3.2. Mean-Shift Algorithm

The mean-shift algorithm (Comaniciu et al. 2003) is a non-parametric density gradient estimator which finds local maxima of a similarity measure between the color histograms (or kernel density estimations) of the model and the candidates in the image. Such algorithm is very fast due to its searching strategy but however it is prone to failure when the target having large motion or whenever occlusion occurred.

In 2009, Bhuvaneswari and Rauf (2009) introduced the use of silhouette oriented feature called edgelet feature for the purpose of detecting human. The discriminative classifiers of objects of a known class were learnt and applied to the video sequence frame by frame. Combined with a soft decision which consisted of a set of detection, it provided the observations used for tracking. For the tracking, mean shift style tracker was used, as well as integrating data association and the use of trajectory. The experimental results had shown that the proposed system had low false alarm rate and achieved a high tracking accuracy.

2.2.3.3. Covariance Matrices Matching

One kind of tracker, called covariance tracker, was used by Porikli et al. (2006) to represent targets using covariance matrices. In particular, it tried to locate the targets in consecutive frames by performing global searching. The covariance tracker searched all the regions and the region with highest similarity to the target model was taken as the estimation result. Wang and Yagi (2008) proposed a tracking strategy that switched between local tracking and global covariance tracking. The switching criteria were determined by the tracking condition, where local tracking was carried out when the target did not have large motion, and covariance tracking was adopted when large motion or occlusions happened to deal with the issue. To speed up the process, Log-Euclidean metrics was used instead of the Riemannian invariant metrics to measure the similarity between covariance matrices.

2.2.3.4. Kalman Filter

Zhou and Aggarwal (2005) presented a robust method for tracking moving objects in outdoor environment which was achieved using feature fusion and multiple cameras. The proposed method integrated spatial position, shape, and color information to perform object blobs tracking. Besides, the authors solved the object occlusion problem by applying an extended Kalman filter which incorporated trajectories obtained from individual cameras. The experiments demonstrated that the overall performance of multiple camera tracking was better than that of a single camera, especially when occlusion occurred. However, using multiple cameras would certainly increase the complexity of the algorithm since the multiple images acquired from different cameras needed to be registered and associated.

2.2.3.5. Optical Flow Tracking

Optical flow algorithms are among the best methods for the estimation of disparity in stereo images and motion in video sequences (Ince and Konrad, 2008). However, one major drawback of optical flow tracker is that it is not capable of dealing with object occlusion. In Ince and Konrad’s (2008) work, they mentioned a three-step object tracking with occlusion areas using optical flow tracking algorithm. First, occlusion-ignorant optical flow was estimated, then occlusion areas were identified using the estimated optical flow, and, finally, the optical flow was corrected using the computed occlusion areas. However, such approach did not permit interaction between optical flow and occlusion estimates. Thus, the authors proposed a variational formulation that jointly computed optical flow and implicitly detected occlusions and extrapolated the optical flow in occlusion areas. The proposed algorithm had shown significant improvement over original and edge-preserving optical flow formulations.

Denman et al. (2009) addressed the weakness of optical flow tracking in which it was prone to errors caused by changing lighting and occlusions. Hence, they proposed an improved optical flow tracking by incorporating motion segmentation results into the tracking process. Motion segmentation results were used to inform the optical flow calculations and ensure that optical flow was only calculated in regions of motion. Note that optical flow was calculated at pixel resolution and tracking of flow vectors was employed to improve performance and detect discontinuities. Each of the tracked objects was identified as a cluster. Overlapping objects were detected based on 4 states i.e. *New, Continuous, Overlap,* and *Ended* which could be determined using the propagated optical flow information. The proposed method was evaluated in terms of accuracy as well as performance. The experimental results had shown that the proposed method performed better.

2.3. Path Detection and Clustering Module

Paths are usually represented by a spatial model that determines the regions or area frequently used by people in the scene. In order to be able to determine the region or area, it is required to have an efficient method to record individual tracks to construct a list of motion patterns which is also known as path clustering. Object trajectories can be assigned to one of only a small number of detected pathways, resulting in significant compression for the logged data. Also, by accumulating data about the motion patterns over a long time period, it is possible to establish a norm of typical movements and this can support the recognition of an unusual movement. There are commonly 2 types of path clustering:-

- Distance Metric
- Clustering strategy

2.3.1. Distance Metric

An approach of clustering similar trajectory would be using the Euclidean Distance Method that is proposed by Fu, Z., Hu, W. and Tan, T.. By getting all the coordinates of the similar trajectory, a simple calculation will be computed and the median will be plotted out. (Fu, Z. et al., 2005) The advantage of using this method lies with the simplicity of the calculation. However, because the calculation is simple, it is also computational extensive and thus it is also a disadvantage of using this method.

Unlike the Euclidean distance, Dynamic Time Warping optimally aligns or “warps" the data points of two time series as shown in Figure 2.2.

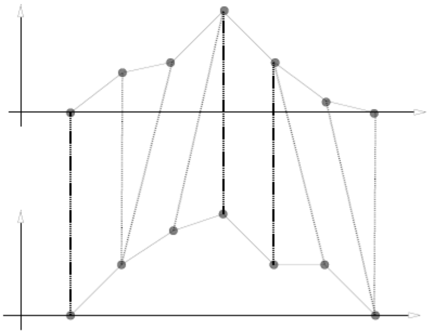


Figure 2.2: Dynamic Time Warping

Dynamic Time Warping (DTW) was initially introduced to recognize spoken words (Sakoe, H. and Chiba, S., 1978), but it has since been applied to a wide range of information retrieval and database problems: handwriting recognition (Bahlmann, C., 2004; Niels, R. and Vuurpijl, L., 2005) signature recognition (Chang, W. and Shin, J., 2007; Faundez-Zanuy, M., 2007), image de-interlacing (Almog, A. et al., 2005), appearance matching for security purposes (Kale, A. et al., 2004), whale vocalization classification (Brown, J. C. et al., 2006), query by humming (Zhu, Y. and Shasha, D., 2003; Jang, J.-S. R. and Lee, H.-R., 2008), classification of motor activities (Muscillo, R. et al., 2007), face localization (Lopez, L. E. M. et al., 2007), chromosome classification (Legrand, B. et al., 2007), shape retrieval (Bartolini, I. et al., 2005; Marzal et al., 2006) and so on.

In surveillance scenes, many applications depend on analyzing motion trajectory, such as path modeling, activity recognition, and anomaly detection. There is a method that is proposed by Keogh, E.J. and Pazzanin, M.J. that is an alignment based distance called the Dynamic Time Warping (DTW). The basic idea behind DTW is to find out the warping path W between two trajectories that minimizes the warping cost. (Keogh, E.J. and Pazzanin, M.J., 2000) Although this method is able to find out the warping path between two trajectories that minimizes the warping cost, it does not however exhibit the robustness with noise.

Beside DTW, several similarity metrics have been proposed including the directed and general Hausdorff distance, Hidden Markov Model-Based Distance, Longest Common Subsequence (LCSS), nonlinear elastic matching distance (Veltkamp, R. C., 2001), and SimilB (Boudraa, A.O. et al.,2008).

The Longest Common Subsequence (LCSS) distance finds the alignment between two sequences that maximize the length of common subsequence. This algorithm is proposed by Vlachos, M., Kollios, G. and

Gunopulos, D.. (Vlachos, M. et al., 2002) Although this method is classified as one of the more efficient method to cluster similar trajectory, it is however geared more towards recognizing shape similarity rather than path trajectory. (Xie, D. et al., 2004)

Bashir, F.I., Khokhar, A.A. and Schonfeld, D. proposed a method called the Principle Components Analysis (PCA) where a trajectory is first represented as a 1-D signal by concatenating the x- and the y-projections. (Bashir, F.I. et al., 2003) Then the signal is converted into the first few PCA coefficients. The trajectory similarity is computed as the Euclidean distance between the PCA coefficients. There is a disadvantage to this method. It is brittle to the distortion and other noise. However, it can cluster similar trajectory in a short amount of time.

Hausdorff Distance is a method that is used by Junejo, I., Javed, O. and Shah, M. to cluster similar trajectory. (Junejo, I. et al., 2004) The method get the longest distance among the two shortest distances between two coordinates. The advantage of this method is that it can get the shortest distance after comparing with all possible possibilities. However, this method is unable to properly recognize the direction of the trajectory. The same path but different direction is the same before this method. However, this method is improved by using a combination of Hausdorff Distance with a slight auto calibration of the Camera. (Imran N. Junejo and Hassan Foroosh, 2009) By calibrating the camera it is possible that clustering the path is made easier and more efficiently.

Porikli, F.M. and Haga, T. mentioned that using Hidden Markov Model-Based Distance is a viable solution to cluster similar trajectory. (Porikli, F.M. and Haga, T., 2004) By comparing the likelihood of the trajectories to their own model, similar trajectory will be clustered together. By using probabilities, it is indeed an efficient method. However, due to limited number of training data, it will most likely suffer from over-fitting.

Veltkamp consider shape as something geometrical and uses shape matching as a distance called nonlinear elastic matching distance (Veltkamp, R. C., 2001). It is quite common to consider it as such although 'shape' is periodically used for a geometrical pattern modulo some transformation group such as similarity transformations (combinations of translations, rotations, and scaling). Shape matching handles shape transformation, and measuring the resemblance with another one, using some similarity measure. Therefore, shape similarity measures are an essential ingredient in shape matching. Although the term similarity is often used, dissimilarity corresponds to the notion of distance where small distance means small dissimilarity and large similarity. However, the algorithm to compute the similarity often depends on the precise measure, which depends on the required properties, which in turn depends on the particular matching problem for the application at hand. Furthermore, this method is more geared towards searching similar image shape rather than similar path shapes.

A new nonlinear similarity measure for Time Series analysis, SimilB, which takes into account the temporal information, is introduced. Using the first and second derivatives of the Time Series, SimilB is able to capture temporal changes and discontinuities of the Time Series. Some new properties of ΨB are presented showing, particularly, that the interaction measure is robust both to time shift and amplitude scale. It is also shown that if the time of the signals is scaled by a factor, the corresponding interaction energy is proportional to that of the original ones. Thus, the time corresponding to the maximum of interaction is unchanged by time scale. However, SimilB is not able to handle very noisy Time Series.

2.3.2. Clustering Strategy

When the distance between two time series forms a metric, such as the Euclidean distance or the Hamming distance, several indexing or search techniques have been proposed.

Neil Johnson and David Hogg present a statistically based model of object trajectories which is learnt from image sequences as a viable solution to cluster similar trajectory and create a path model. (Neil Johnson and David Hogg, 2000) Trajectory data is supplied by a tracker using Active Shape Models, from which a model of the distribution of typical trajectories is learnt. From the sequence of the image shown during the training stage, the trajectories of the tracked individual can be obtained. However, because the amount of data is too large, a large storage is needed.

Dimitrios Makris and Tim Ellis develop a spatial model to represent the routes in an image. (Dimitrios Makris and Tim Ellis, 2002) A trajectory is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function; otherwise a new route is created for the new trajectory. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person.

Recent advances in object tracking also made it possible to do path in a large trajectories set. Cluster analysis is a common clustering method. For similarity based clustering, a key issue is how to measure the similarity between two trajectories. (Zhang Zhang et al., 2006) However, such clustering can only be done after going through a training period. Without going through the training stage, it is impossible to cluster path that has yet to exist. In other words, the “normal path” that can be clustered must exist so that we can determine what path can be taken while what path can’t be taken.

One clustering approach that is proposed by Ujjwal Das Gupta, Vinary Menon and Uday Babbar is to assume a Gaussian Mixture Model and implementing the X-Means algorithm. In this model, the input data is assumed to have been generated by selecting any one of the K Gaussian distributions. (Ujjwal Das Gupta et al., 2010) Each cluster is thus represented by a single distribution. The major problem that is faced by using the X-Means is that is assumes an identical spherical Gaussian of the data and thus it tends to overfit data in elliptical clusters. The advantages of using this algorithm is that can optimally detect the clusters for a chosen value of K.

However, even assuming that there is a metric, Weber et al. have shown that the performance of any indexing scheme degrades to that of a sequential scan, when there are more than a few dimensions (Weber, R. et al., 1998). Otherwise, when the distance is not a metric or that the number of dimensions is too large, bounding techniques will be chosen such as the Generic multimedia object indexing (GEMINI) (Faloutsos, C., 1996). False positives will be discarded by computing a lower bound.

Boundary-based lower-bound functions sometimes outperform LB_Keogh (Zhou, M. and Wong, M. H., 2007). We can also quantize (Vega-Lopez, I. F. and Moon, B., 2006) the time series. Sakurai et al. (2005) have shown that retrieval under the DTW can be faster by mixing progressively finer resolution and by applying early abandoning (Wei, L. et al., 2005) to the dynamic programming computation. All the methods above are more geared towards Shape Time Series or Forecasting data sets such as stock price or currency exchange rates.

2.4. Suspicious Path Detection Module

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform Closed Circuit Television (CCTV) systems from ‘post-mortem tools’ into the forefront of crime-fighting tools (Sang-Hyun, O. et al., 2005). The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel. The problems posed in detecting suspicious activities are not new. Some problems were also investigated in the area of network-based intrusion detection systems; such systems detect an intrusion by examining packets in the network. Approaches in intrusion detection are classified into 2 types:-

- Anomaly detection model
- Contextual information.

2.4.1. Anomaly Detection Model

The misuse detection model attempts to create attack profiles. An intrusion is detected when there are patterns matched with the created profiles. This model works quite efficiently for the attacks that has been predefined previously, however, it will fail to detect attacks that are not defined yet. Thus, in order to overcome this problem, the anomaly detection model was proposed. The model creates a long-term usage profile that represents the common users' activities. An attack is detected when the short term such as the current user patterns profile deviates too far from the long-term profile. In the surveillance field, both of these models are implemented and researched. The latter approach, anomaly detection model, is becoming more popular since it is able to handle unknown profiles.

There are some approaches which use the anomaly detection model (Vaswani, N. et al., 2003; Hua, Z. et al., 2004; Boiman, O. and Irani, M., 2005; Makris, D. and Ellis, T., 2005; Piciarelli, C. and Foresti, G. L., 2006; Yue, Z. et al., 2007). For example, Vaswani et al use shape feature to calculate the common shape of walking paths in airport scenario (Vaswani, N. et al., 2003). Any walking path deviated much from the common shape is labeled as suspicious. The feature shape is very attractive as it allows the system to calculate the average shape. However, averaging features is not always applicable to represent the common patterns. For example, if the speed of a person is averaged, it would be hard to distinguish between the running and walking actions. Measuring the deviation of a behavior pattern to the others could be done in different ways. For example, one may describe the deviation in terms of whether behavior can be constructed from the common database or not (Hua, Z. et al., 2004; Boiman, O. and Irani, M., 2005). Another way is to redefine the problem such that behavior deviates from normal when it cannot be classified into one of the common behavior categories (Yue, Z. et al., 2007). These approaches, however, have to have the database contains all possible normal behavior to be available. This becomes a big issue in the real-life scenarios since it is difficult to define all possible normal behaviors. To illustrate the difficulty one may look at what Jianbo et al coined (Hua, Z. et al., 2004). They pointed out that the number of suspicious behavior types are less than the normal ones. In addition, defining all possible suspicious behavior is posed as difficult problem. So, defining all possible normal behavior is even a more difficult problem.

Several other approaches from the aforementioned works have been proposed (Makris, D. and Ellis, T., 2005; Piciarelli, C. and Foresti, G. L., 2006). A variant of the leader-and follower algorithm is presented in (Duda, R. O. et al., 2000) to create the normal behavior profiles. These approaches are able to continuously learn not only from the predefined training sets, but also during the operational time. Nevertheless, some required parameters (e.g. Initial variance, the notion of closeness between two different behavior types) which have to be set in advance, are difficult to set. From previous approaches, although each author gives advancement in this area, but there is a common issue which needs to be addressed. A normal behavior model is needed to detect suspicious behavior. Most methods generate normal behavior model from training sets. However, but, it is nearly impossible to include all possible behaviors in the training sets to create a normal behavior model. In other words, the training sets always have insufficient possible types of normal behavior. This means that any method that relies only on training sets to have over fitting issue. One way to overcome this problem is to give the method which is able to learn not only from training sets, but also during operational time. A behavior pattern that initially might be considered as an anomaly could become normal in long term observations (Morris, B. T. and Trivedi, M. M., 2008). The methods used by Piciarelli et al and Makris et al follow this line of thinking. Nevertheless, defining the right values of the required parameters is still problematic.

Hidden Markov Model is also another method to detect suspicious behavior. Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato and Yozo Fujino propose a method detects anomalies and classifies into several behavior patterns based on probabilistic distances between HMMs. (Naohiko Suzuki et al., 2007) The framework is shown below:-

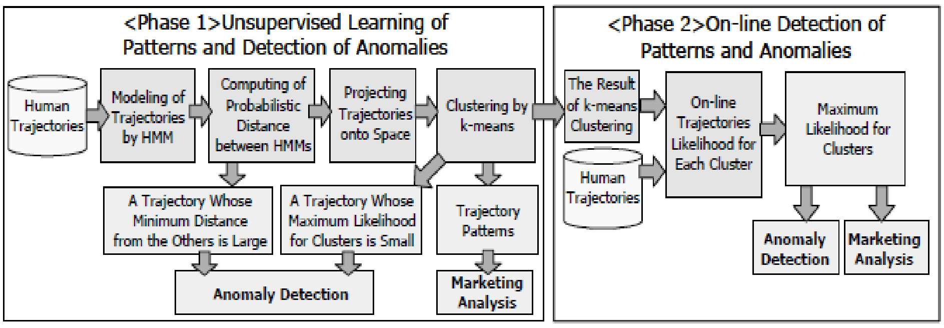

<Phase 1>Unsupervised Learning of Patterns and Detection of Anomalies
Human Trajectories
Modeling of Trajectories by HMM
Computing of Probabilistic Distance between HMMs
Projecting Trajectories onto Space
Clustering by k-means
A Trajectory Whose Minimum Distance from the Others is Large
A Trajectory Whose Maximum Likelihood for Clusters is Small
Trajectory Patterns
Anomaly Detection
Marketing Analysis
<Phase 2>On-line Detection of Patterns and Anomalies
The Result of k-means Clustering
Human Trajectories
On-line Trajectories Likelihood for Each Cluster
Maximum Likelihood for Clusters
Anomaly Detection
Marketing Analysis

Figure 2.3: Reference from “Learning Motion Patterns and Anomaly Detection by Human Trajectory analysis”, IEEE 2007

They have tested the method using an online dataset and also implement unsupervised training. The advantages of using this method is that the system that implements this method can be used anywhere without any prior knowledge of the area.

Another way to detect suspicious behavior would be the Mixture of Von Mises method that was proposed by Simone Calderara, Rita Cucchiara and Andrea Prati. They decided to model the behavior in terms of the trajectory of movement, modeled as a sequence of directions computed as the angle between two consecutive positions and that allows modeling robustly trajectories even in presence of noisy observations. (Simone Calderara et al., 2007) The disadvantage of this method is that it would be hard to create a training set. On the other hand, this method is both intuitive and discriminative enough to detect suspicious behavior efficiently. In another work of the same author, they made an improvement. By including the Expectation Maximization, they can now implement the method using an unsupervised learning. The previous disadvantage of no training set is available is now solved. (Simone Calderara, Rita Cucchiara and Andrea Prati, 2008)

2.4.2. Contextual Information

There can be a different perspective on the problem. It is shown that contextual information is under-utilized in the aforementioned methods. Recently, there has been some other interesting works which successfully show the benefits of using contextual information in vision systems.

Gupta et al (Gupta, A. and Davis, L. S., 2007) show that contextual information can be discovered from interaction between humans and the objects being manipulated. An object can be better recognized when one knows how the object is being manipulated. Conversely, one may expect how the person’s action will be performed when one knows what type of the object the person will manipulate. Furthermore, they also showed that manipulation actions can be recognized by looking at the object’s reactions. For instance, one may expect a drastic change in image illumination when a person pushes a pocket-sized camera button which triggers a flash light to come out. The motion when the person pushes the button may not be observable because it is too subtle to detect, however the flash lights coming from the pocket-sized camera can be used to infer what the person has done.

Morency et al (2006) use conversational script which is extracted by using a speech recognition technique as contextual information to help the vision systems understand head gesture behaviors. For example, when there is a question mark at the end of a sentence, it is more likely that a nod or head shake gestures will be observed. They showed that by using this information, the system performed better.

Katz et al (2007) explored the usage of contextual information to search for an object in an image. By knowing the object to object relationship, their system could locate an object in an image more accurately. For instance, detecting a computer monitor could be a hard problem; nevertheless one may use contextual information to solve this problem. In this case, one may use the easier-to-detect objects which usually appear altogether with monitor such as keyboard or mouse to help with detecting it. Spatial relationships also can be exploited such as a keyboard most of the time is located below a monitor.

Freer, J A et al. (1997) proposed an image processing based approach for the automatic identification of moving objects within monochrome video images. The tracking algorithm is furnished with scene dependent knowledge which takes the form of a coded bitmap representation of the original scene being viewed. This bitmap representation is segmented into areas which have significance in terms of the physical security of that area. Each area is assigned a risk factor which relates to the security risk posed by the presence of persons within that area. Should a pixel escalated the threshold set by user, the human operator will be alerted and this is considered as suspicious. The advantage of using such technique is that the method is simple to implement and it can be implemented at any terrain. However, the disadvantage is that the terrain must be known to the system beforehand through a supervised training.

In another work by Chun Che Fung and Nicholas Jerrat, they have pointed out a serious technical difficulty that must be solved in order for a detection system to be workable and usable. VMD systems must differentiate between an intruder and environmental conditions such as rain, wind, fog, birds, animals, and lightning effects. (Chun Che Fung and Nicholas Jerrat, 2000) While some are easy to differentiate, others have proven to be quite difficult and pose as a great challenge to overcome. Thus, in order to solve this problem, the motion detection must be improved so that only the right object (intruder) will be detected. For a motion detection algorithm, the steps that need to be taken are shown as follow:-

- Grab the current video image
- Compare to the previous image and to determine whether there is a significant motion
- Identify the units causing the motion if any
- Pass each unit through a series of tests to eliminate false alarms
- Trigger response if a genuine intruder is detected
- Repeat the steps above

The current images are always being compared with the previous images so that small changes such as the global lighting effects can be integrated as time progresses without triggering a false alarm. The advantage of doing so is that the storage requirements will be reduced and the impact of random pixel size will change.

Another method would was proposed by a paper that is written by Simon Fong and Zhuang Yan. Their methods are referring to user activities and actions that are reflected by their movement. (Simon Fong and Zhuang Yan, 2007) They argue that by connecting multiple access reference points, they can gain a better understanding of the user's behavior than a single entry validation verdict such as access "Granted" or "Denied". Their solution adopted SRI's NIDES technique, for implementing a user's profile that has a set of statistical measures. The advantage of using this method is that the user's patterns can be learnt and the system is able to tell whether a new trail is normal or suspicious.

Daniel Barbara and Maurizio Filippone have proposed a design of a novel technique that will detect suspicious behavior. Their implementation uses standard techniques for foreground and background object detection in images. (Daniel Barbara and Maurizio Filippone, 2008) First, they will take an image of a foreground. Next, any object that is not present in the foreground and yet has appeared on video is considered suspicious. This method is useful in detecting suspicious behavior or suspicious object. However, the downside of this method is that they must first define a meaningful set of features, and a proper distance measure between images or else this method is not applicable.

Detecting uncommon trajectories is also a feasible method. This method is proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda. They used a dataset from CAVIAR 1 which are available publicly. The example of images is shown below:-

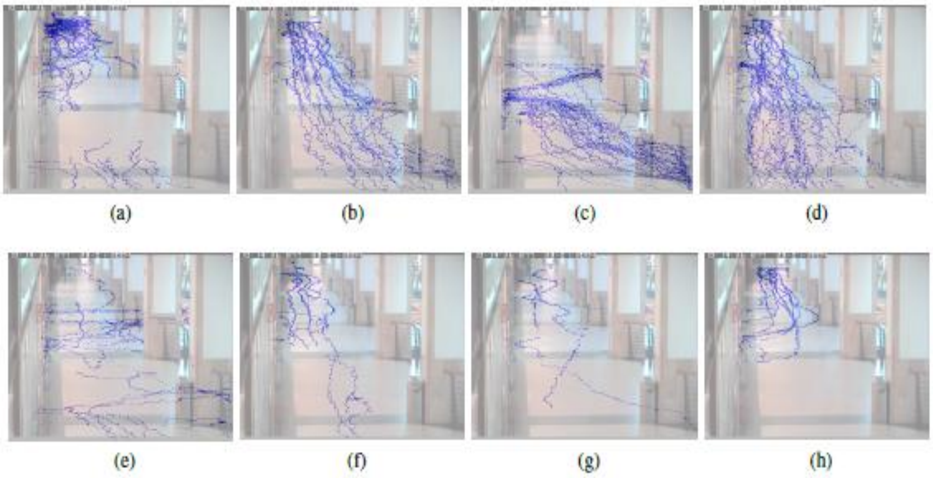


Figure 2.4: Reference from “Detecting Uncommon Trajectories”, IEEE, 2008

For the purpose of the experiment, they only stick to the basic lower level processing and focus more on the tracking and detection of suspicious behavior. They will first trace the people's position in each image sequence and the person's trajectory will be formed up from these coordinates. Then, from the trajectory, they will then group them into clusters of similar trajectories. Those trajectories that do not fall into any of the clusters will be deemed suspicious. (Arnold Wiliem et al., 2008) The disadvantages of using this heuristic method is that it will not be very effective when dealing with the real life situation as all the suspicious behavior pattern that is used is mainly created for experiment purposes.

The solution that were proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda explore two contextual features, such as type of behavior as well as the commonality index of each behavior type as contextual information, to detect suspicious behavior. This solution uses shape feature to calculate the common shape of walking paths in airport scenario. Any walking path deviated much from the common shape is labeled as suspicious. (Arnold Wiliem et al., 2009) The advantage of this method is that it will be more adaptable to the unseen patterns and it does not need the training sets to include all possible types of normal behaviors. However, this model will only work well provided that the database for all kind of possible attack is available. Thus, newly created attacks will not be detected. Yi Feng Zhou and Jim P.Y. Lee however show that this problem can be solved. (Yi Feng Zhou and Jim P.Y. Lee, 2010) The method proposed by them is a modified expectation maximization algorithm for maximum likelihood direction of arrival estimation can be used for real life update and thus newly created attacks that cannot be detected by Arnold and his partner’s work but can be detected by Yi Feng Zhou and Jim P.Y. lee’s work.

2.5. Conclusion

In conclusion, after reviewing all the aforementioned papers above, a method to detect and cluster similar path will be used so that it can further help with the suspicious behavior detection based on trajectory based activity analysis. Experiment to further test the method that will be used will be done on data sets that will be taken either online or self-captured and further improvement will be done should there be any need of it.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 Introduction

This chapter will discuss how the system is being implemented. A detailed discussion of each sub-module that exists in this research work, the structure of the system and the analysis of the hardware and software requirements is also included in this chapter. This chapter will show the system dependency, definitions, assumptions, system limitation, the adaptation of the system in different environments, etc. In addition, this chapter will also include the variables used and the discussion about the system's testing environment.

The system architecture of the system is drawn and shown in the next section. The Security Alert through Path Analysis System is basically made up of three modules. The modules are:-

- Human Detection
- Path Detection and Clustering
- Suspicious Path Detection

Module 1 is responsible for detecting human that appears in the scene and tracking their positions. Module 2 basically is in charge of getting the path taken by the human that is detected in Module 1, group it into the correct cluster, and store the cluster in the database in order to form a generic topographical scene description which is passed to Module 3 for further analysis. In this chapter, we will explain the concept that we have in extracting the required information and also the method to alert security through trajectory based activity analysis.

For the test subject, the human that is tracked in the scene must not be occluded. Any human that is passing by another human with part of the body being blocked from vision is considered as occluded as shown in Figure 3.1.

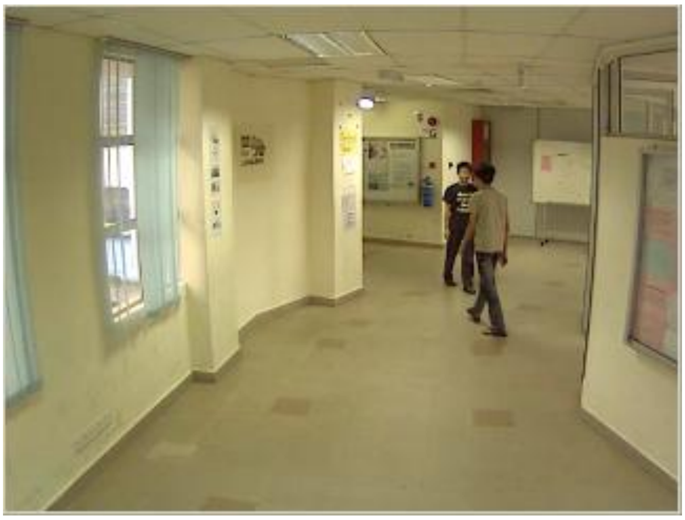


Figure 3.1: Human Occlusion

If there is something blocking the human while being tracked, this human is also considered as occluded as shown in Figure 3.2. The system is designed for indoor use and will be tested in an indoor environment.

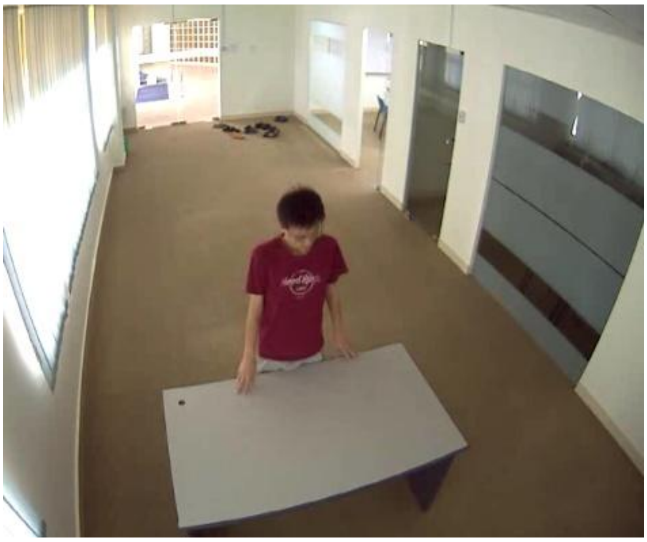


Figure 3.2: Object Occlusion

3.2 Diagrammatic Representation

Security Alert through Path Analysis

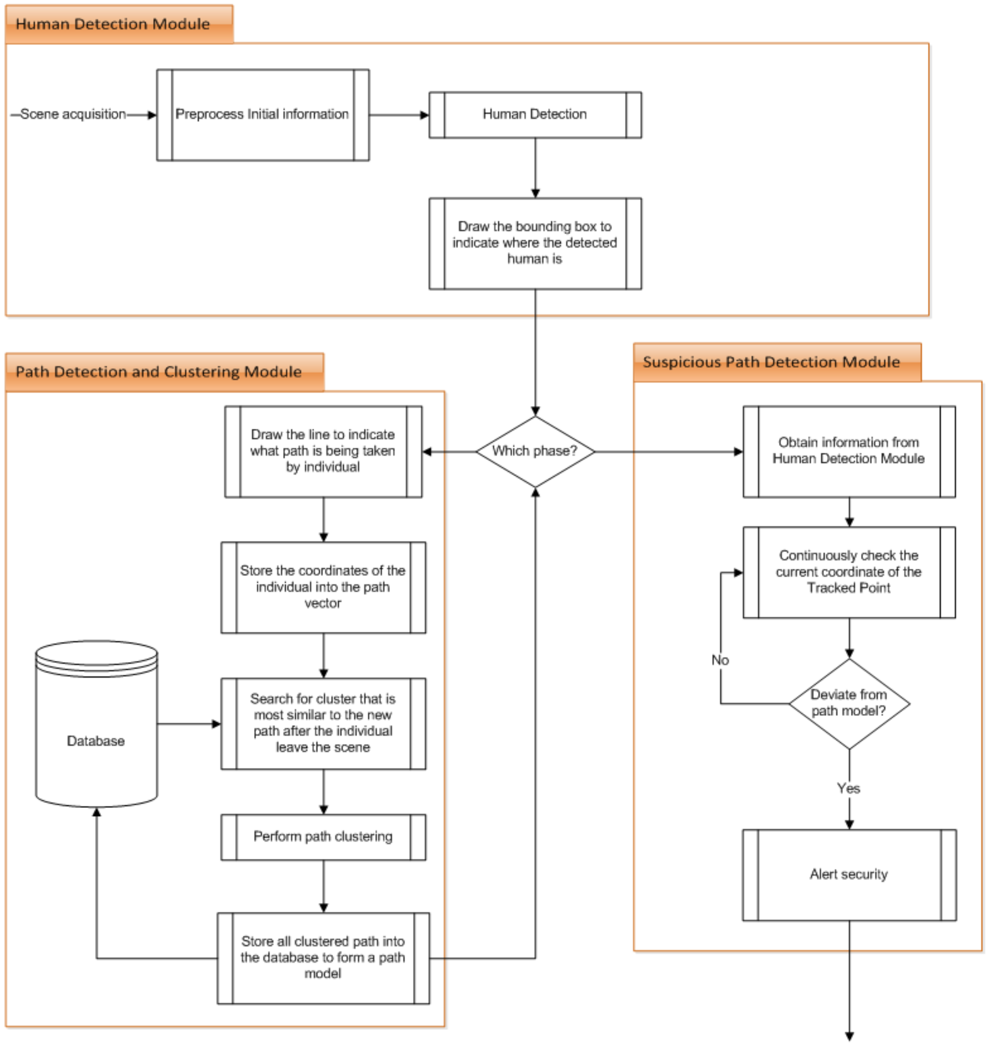

Human Detection Module
Scene acquisition
Preprocess Initial information
Human Detection
Draw the bounding box to indicate where the detected human is
Path Detection and Clustering Module
Draw the line to indicate what path is being taken by individual
Store the coordinates of the individual into the path vector
Database
Search for cluster that is most similar to the new path after the individual leave the scene
Perform path clustering
Store all clustered path into the database to form a path model
Which phase?
Suspicious Path Detection Module
Obtain information from Human Detection Module
Continuously check the current coordinate of the Tracked Point
No
Deviate from path model?
Yes
Alert security

Figure 3.3: System Architecture

The Figure above shows the system architecture of the system. As shown above, the system consists of 3 modules. The proposed system's title is "Security Alert through Path Analysis". This proposed system is aimed for the automated security alert in a particular closed-circuit television monitored area. For example, whenever an object enters the region of the monitored scene, the system will begin to analyze the situation and monitor the individuals in the video; once the moving individual are monitored, the system will further analyze the path taken by the individual. If the detected person is walking with an uncommon style or the path taken is relatively uncommon, he or she is deemed as suspicious.

This system is divided into three modules which are “Human Detection”, “Path Detection and Clustering Module” and “Suspicious Path Detection Module”. Each module has their own functionality that is designed for different purposes that contributes to the overall objective.

This system will begin with Module 1 – Human Detection Module. Human Detection is the fundamental part of the whole proposed system. The main task of this function is to identify whether an object appeared in the monitored scene is a human. This is because human is the only object that will cause security threats and worth of human’s scrutiny. The image captured will undergo a series of preprocessing techniques to filter out unwanted information such as noises and non-relevant images such as areas that are not accessible and will not affect the overall outcome of the system. This function is also combined with the use of a masking technique, which helps filter out the uninterested regions, i.e. the wall, some inaccessible areas, etc. Such technique also helps preventing the system from wasting resources on the processing of unnecessary computations. If the object is classified as human, the Tracking Point (TP) of the person will be computed and passed to the Module 2 – Path Detection and Clustering Module.

Module 2 – Path Detection and Clustering Module is mainly responsible for detecting the path that is taken by the individual captured in the video and cluster all similar trajectories together so that it will further improve and help in the analysis of the path for suspicious behavior detection. This module is triggered after a human is detected and the Tracking Point (TP) of the human is taken. By using the Tracking Point (TP) of the human, a path can be drawn to indicate which path is taken by that individual and by knowing which path is taken; it can then be clustered to the nearest and most similar cluster for further trajectory based activity analysis.

The third module, Suspicious Path Detection Module, is responsible for analyzing the path by monitoring the activity of individual that is captured in the video. The system will then determine the status of an individual based on the analysis of the path taken by the individual. Finally, through Module 2, a normal path will be determined and should any individual deviate from this normal path the security personal will be alerted and further security actions will be taken.

A summary of the functionality of each sub-system and its modules is indicated as in Table 3.1.

Table 3.1: System Model Description

Process Code	Function	Description
Human Detection Module		Responsible for detecting the motion region and identify human.
HD1	Capture first frame as the background image and store it. Subtract foreground image with the background image. Perform pre-processing on the result that is obtained from the previous operation.	Once the camera starts recording, the first frame will be stored as background image for the system. All frames excluding the first frame are foregrounds. The foreground image will be subtracted with the background image. Perform pre-processing on the image to obtain a better image for the usage of human detection.
HD 2	Human Detection	Calculate certain features to be used in the identification of the human posture and action.
HD 3	Draw the bounding box to indicate where the detected human is	Draw the bounding box so that the Tracking Point (TP) can be obtained through a simple calculation.
Path Detection and Clustering Module		Responsible for detecting the path that is taken by the individual that is being monitored in the video and cluster all similar trajectories together.
PC 1	Draw the line to indicate what path is being taken by individual	Indicate which path is taken by the individual that is being monitored.
PC 2	Store the coordinates of the individual into the path vector	All the points that is obtain will be stored for path clustering usage.
PC 3	Search for cluster that is most similar to the new path after the individual leave the scene	Search through all available cluster to find the cluster that is most similar with the new path.

PC 4	Perform path clustering	Cluster the path taken to a similar clustered path for further analysis.
PC 5	Store all clustered path into the database to form a generic topographical scene description	All clustered path will be stored in the database to form a generic topographical scene description.
Suspicious Path Detection Module		Responsible for detecting individual that takes the unusual path
PA 1	Detect Human	Human that appear in the scene will be detected.
PA 2	Track the human and get the Tracking Point of the tracked human	The detected human will be tracked and the Tracking Point of the human will be stored.
PA 3	Continuously check the current coordinate of the Tracking Point	Individuals that are being monitored will have their status determined based on current Tracking Point's coordinate
PA 4	Alert security	The security guards will be alerted and the individual that triggered the security alert will be monitored by the security guards.

3.3 Proposed Methodology

3.3.1 Module 1 - Human Detection

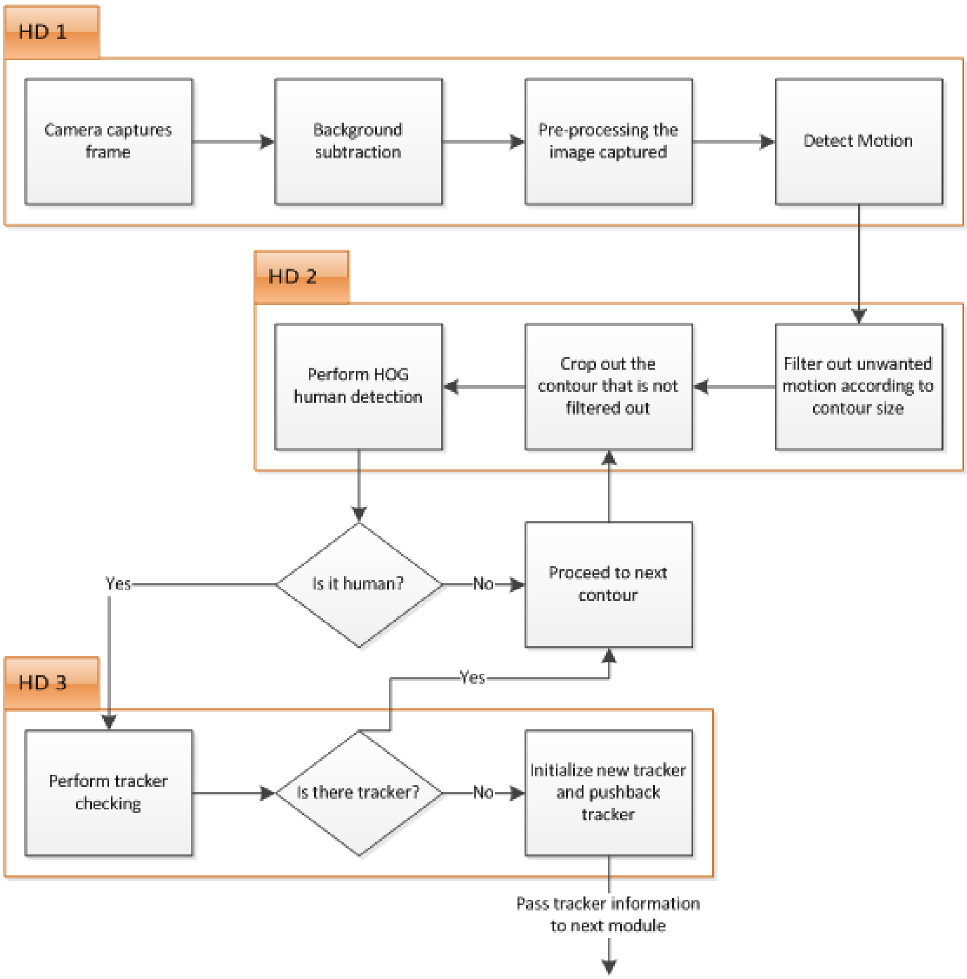

HD 1
Camera captures frame
Background subtraction
Pre-processing the image captured
Detect Motion
HD 2
Perform HOG human detection
Crop out the contour that is not filtered out
Filter out unwanted motion according to contour size
Is it human?
Yes
No
Proceed to next contour
Yes
HD 3
Perform tracker checking
Is there tracker?
No
Initialize new tracker and pushback tracker
Pass tracker information to next module

Figure 3.4: Module 1 Structure Diagram

Figure 3.4 shows the breakdown of the process found in the Human Detection Module. The main purpose of this sub-system is to determine whether the moving object is a human.

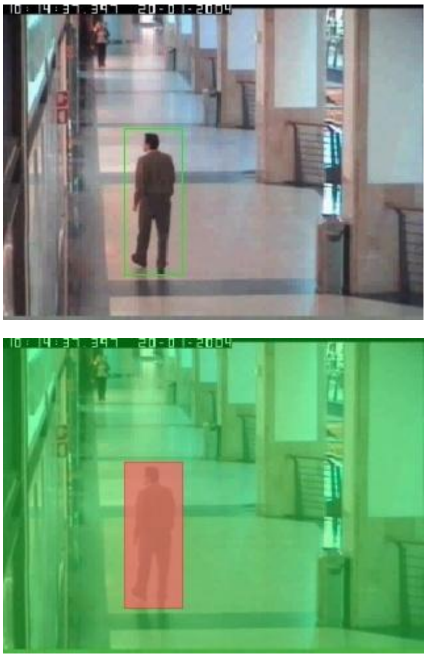
3.3.1.1. Preprocessing Initial Information (HD 1)

To start off, there will be only one camera being used for this system. For better accuracy, the camera will be installed in a fixed position and not moving. The moment the system starts, this module will start detecting for humans that may appear in the scene. Background model is a crucial part in Human Detection module. It represents the background information of a particular scene and will affect greatly the end results for this module.

In this project, adaptive median (or called as running average) background model update method is chosen. It has the advantages of simplicity, fast in performance, and also the adaptability in background motion. It slowly learns the background according to a defined learning rate. Besides, adaptive median background model update method does not require training stage prior to the detection. Equation 1 illustrates the background update process where B denotes the background frame, C represents the current frame and α as learning rate. In this case, the learning rate α is defined as $0.03\ (3\%)$.

$$B_{i+1} = \alpha \times C_i + (1 - \alpha) \times B_i \text{ .... (1)}$$

However, such background update method may have the disadvantage of adapting moving objects into the background model as well. Thus, a checking process named Selective Area Background Model Update is implemented. It performs segmentation to the scene, thus updates the background model only on the selective areas which does not contain the detected moving objects. Figure 3.5 illustrates the process of selective area background update.



Human is detected and a tracking box being located on it (green rectangle).

Green-colored area indicates the area that will be updated in background model update process; red-colored area will not be updated in the background model update process.

Figure 3.5: Selective area background model updates process

Since this process requires the detection result from human detection, thus the background model update process is scheduled at the last part of the system. The similar procedures are performed on the inverted background model in parallel. Both original background model and inverted background model will be used for the foreground segmentation part, respectively.

After the background is obtained, a series of pre-processing will be carried out in order to improve the human detection accuracy. Firstly, both background and foreground images will have its color converted into grayscale. Next, the system needs to determine whether there are any moving objects in the scene. For a better result, motion in a scene is best to be detected in a non RGB image. In order to do so, the system will get the absolute difference between the background image and the foreground image. So, any moving objects in the scene will be plotted as a white blob as shown in Figure 3.6.

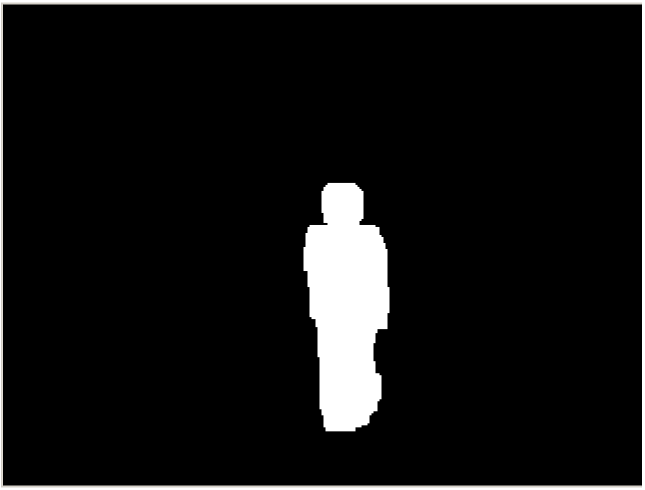


Figure 3.6: White Blob Moving Object

This method of background subtraction is known as the static background subtraction where the background image is a static image instead of a dynamic background where the background image is always updated. The reason static background subtraction is used is because information about the background can be retained and obtained more easily.

By getting the absolute difference between the 2 images, there will definitely be some noises that appear due to the quality of the image taken by the camera. Thus, the noises will then be eliminated by truncating the resulting image from the absolute difference and also binarizing the images so that it will become black and white in color instead of grayscale. The reason for binarizing the image is so that the moving object will be more obvious in a binary color image rather than a grayscale image. Finally, not all noises can be eliminated just by truncating and binarizing the image. Thus, a further noise elimination process is done. In order to get rid of the Gaussian Noise, the binarized image will be eroded follow by a dilation process.

Once the image has been processed through the pre-processing phase, motion detection in the scene will then commence. Should there be any white blob appear in the scene, it indicates that there are motions in the scene and these motions (also known as contours) needs to be checked whether they are human. However, before any human detection can be done, a size filter needs to be done since there might be a lot of motion detected by the camera as lighting might cause some shadow moving or the flickering lights on the window blinds will cause motions. Logically speaking, any contour that is too small will definitely not be human and can be ignored. Thus, the system will filter out any contour that is smaller than 48 pixels in height and 24 pixels in width.

3.3.1.2. Human Detection (HD 2)

After the contours have been filtered, the remaining contours will be passed into the human detection function. However, passing in a 320 pixel x 240 pixel image into the human detection will cause the system to slow down in term of processing speed. This is due to the large number of pixels that needs to be processed in the human detection function. Therefore, in order to speed up the processing time, instead of passing the whole image into the human detection function, only the contour will be passed in. As soon as the filtration of the contour is done, the remaining contour will be cropped out. The area around the contour that is cropped out will be the Region of Interest (ROI).

With the Region of Interest (ROI) determined and the contour is cropped out, the human detection function will then try to determine whether the cropped contour is a human or not. The method that the system is going to use in order to detect human is called Histograms of Gradients (HOG) (Navneet Dalal and Bill Triggs, 2005). HOG is a feature descriptor that will be used to detect human within the scene. This feature descriptor will then be used with a classifier known as Support Vector Machine (SVM) classifier (Vladmir N. Vapnik, 1995).

This classifier has been trained by using datasets from INRIA, MIT and some self-captured datasets. The number of datasets that is used to train the new classifier is 14,713 images from both positive and negative datasets. The number of positive datasets is 6425 images with 4396 images taken from INRIA and MIT while the remaining 2029 images were self-captured. The number of negative datasets is 8288 images. These images are taken from INRIA. For positive images, only images where the whole body of a human is detected will be accepted. Partial detected human dataset will be rejected and will not be accepted in both positive and negative datasets. This is to increase the accuracy of the SVM classifier to detect human.

The human detection function will then use the newly trained SVM classifier to determine whether the cropped contour is human or not. First it will create a sliding window with the size of 24 pixels in height and 16 pixels in width. The sliding window will always start at the top-left of the contour and this window will stride across the cropped contour in an interval of 8 pixels to the right and 16 pixels to the bottom. After the sliding window has reached the bottom-right of the contour, the sliding window will increase the current window size by 5%. The window size will be increased after each time the sliding window reaches the bottom-right of the contour until the window size is either the same size as the contour or larger than the contour.

When the sliding window stops increasing the window size, this means that the human detecting function has come to an end and has reached a conclusion as to whether the contour is human or not. If the SVM has determined that the contour is not human, then it will drop this contour and proceed to the next contour. However, if the contour is determined as human, then this contour will then be passed on to the human tracking function.

3.3.1.3. Draw the bounding box to indicate where the detected human is (HD 3)

Before continuing into the human tracking function, the contour will then be checked for the existence of tracker. If this contour is already being tracked there will definitely exist a tracker that is currently tracking this contour. If there exists a tracker, then this contour will be dropped and proceed to the next contour. However, if there is no tracker in this contour, then the human tracking function will commence. In order to track the human, a new tracker must be initialized for the specific detected human. Each tracker will only track one human and all trackers will be stored in a vector called tracker1. Firstly, once the human detection function has determined that the contour is human, the human tracking function is going to pushback a new tracker into the tracker1 vector.

There are 2 types of tracking functions; the first one is used when there are new humans introduced into the scene while the second one is used for updating the status and position of the existing tracker. Both have the same function names but have different parameters and a slight different internal function. If the human that is detected is not an existing human but a newly introduced human, then the first tracking function will be used.

Thus, the first thing that the tracking function will do is to initialize the Track Point for this human. Track Points are points that are plotted at the edge of any moving object as shown in Figure 3.7.

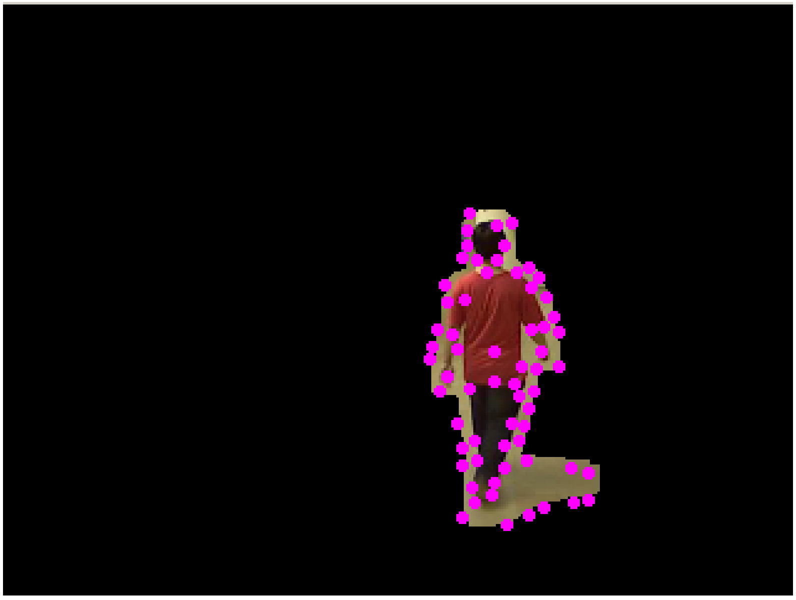


Figure 3.7: Track Point Plotting

All the Track Points that are plotted will be stored in a 2D vector. The number of points that is used to plot the Track Points will be stored under an integer variable called count. However, the Track Points that are stored is not at the correct position. In order to fix that, all the points in the vector must undergo a transformation by adding the coordinates of all the points in the vector with the coordinates of the contour's top-left coordinate. With this coordinate included in the calculation, all the Track Point will then be accurately positioned within the contour.

With the Track Points, the human can now be tracked continuously until the human leaves the scene by calculating optical flow of the Track Points. Before the beginning of the next frame, the Track Points stored in the initial vector will be transferred into a second vector. This vector will be storing all the previous Track Point's coordinates and the Track Points in initial vector will be overwritten with the updated Track Points that is obtained by calculating the optical flow. For a better result of calculating the optical flow, a minimum number of Track Points must exist in the tracked human. Therefore, the number of Track Points in the contour will be maintained. If the number of Track Points in the contour is less than 5 (threshold for the minimum number of Track Points in the contour), all the Track Points will be reinitialized to reach the minimum amount of Track Points needed.

With the Track Points coordinate, the contour's position can be obtained by getting the mean distance of all the Track Point's coordinate. Only the Track Points that is plotted in the contour will be included in the calculations. All Track Points that are plotted outside the contour will be ignored. Though the coordinate has been obtained, the coordinate is the mid-point of the contour. The correct position for the tracker's coordinate that need to be obtained is located at the feet of the human. Therefore, with the obtained coordinate, 40% of the contour's height will be added to the Y- coordinate.

As mentioned above, the contour will first be checked to see whether there exists a tracker currently tracking this contour. If there is a tracker, then the tracker for the specific contour must undergo another checking to make sure that this tracker is the same tracker that is supposed to track the specific contour. This can be done by calculating the distance of the tracker with all existing contours in the scene. If the distance is not greater than a specific threshold, then this tracker is supposed to track this contour. The logic behind this reasoning is that the contour with the least distance to a tracker indicates that this tracker is the tracker that is supposed to track the said contour. With the tracker determined, the next step would be updating the tracker’s information by using the second type of Tracking function.

With all the Track Points in the contour, there will be some instances where a small number of Track Points that are plotted in the contour will remain in the same coordinate without any possibility of moving to another coordinate even when the contour has already left the specific area. These cases will only happen when the background’s color is similar to the detected human’s color or the background has a deeper color than the human’s color. Thus, in order to track a contour, relying only on optical flow update might not be so accurate. To solve this problem, a combination of contour and optical flow update is introduced.

A normal update will only rely on the calculation down by the optical flow in order to determine the new coordinate of the tracker. However, with the instances mentioned above, this method will cause the Track Point to remain in a certain coordinate indefinitely. Thus, a contour update must be included as well in order to allow the coordinate that is stuck to update normally too and not to remain in a certain coordinate indefinitely and the coordinate will then be passed on to the next module for path detection and clustering. A contour update is done by updating the current contour coordinate should the number of track points on the tracked contour is below the minimum value.

3.3.2 Module 2 Path Detection and Clustering

Module 2 will be executed only during the training process and this module’s main purpose is to obtain the path that is taken by the tracked human and cluster them together with the correct cluster that is stored in the database to form a Generic topographical scene description. After the Human Detection Module is executed, a TP is passed into the Path Detection and Clustering Module. This TP is the input of this module and through this TP we can clearly draw a path that is taken by the monitored individual. The TP that is obtained is illustrated below.

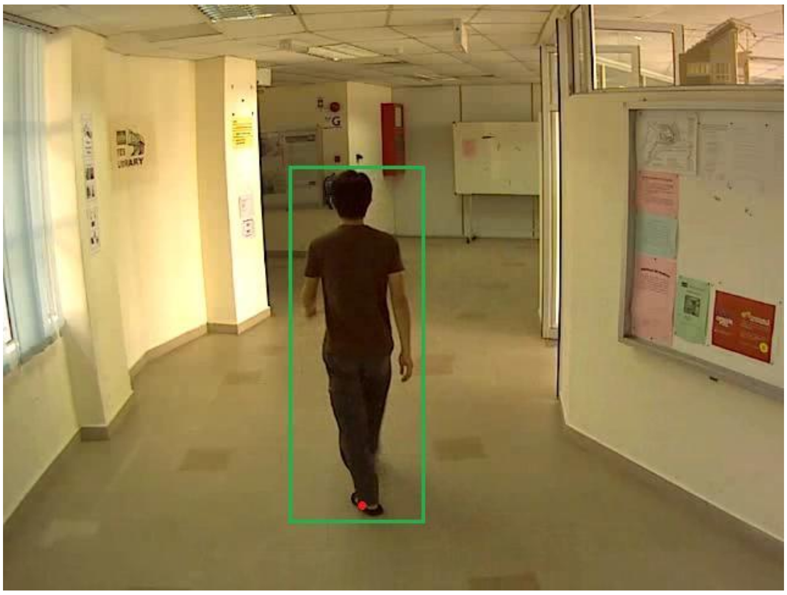


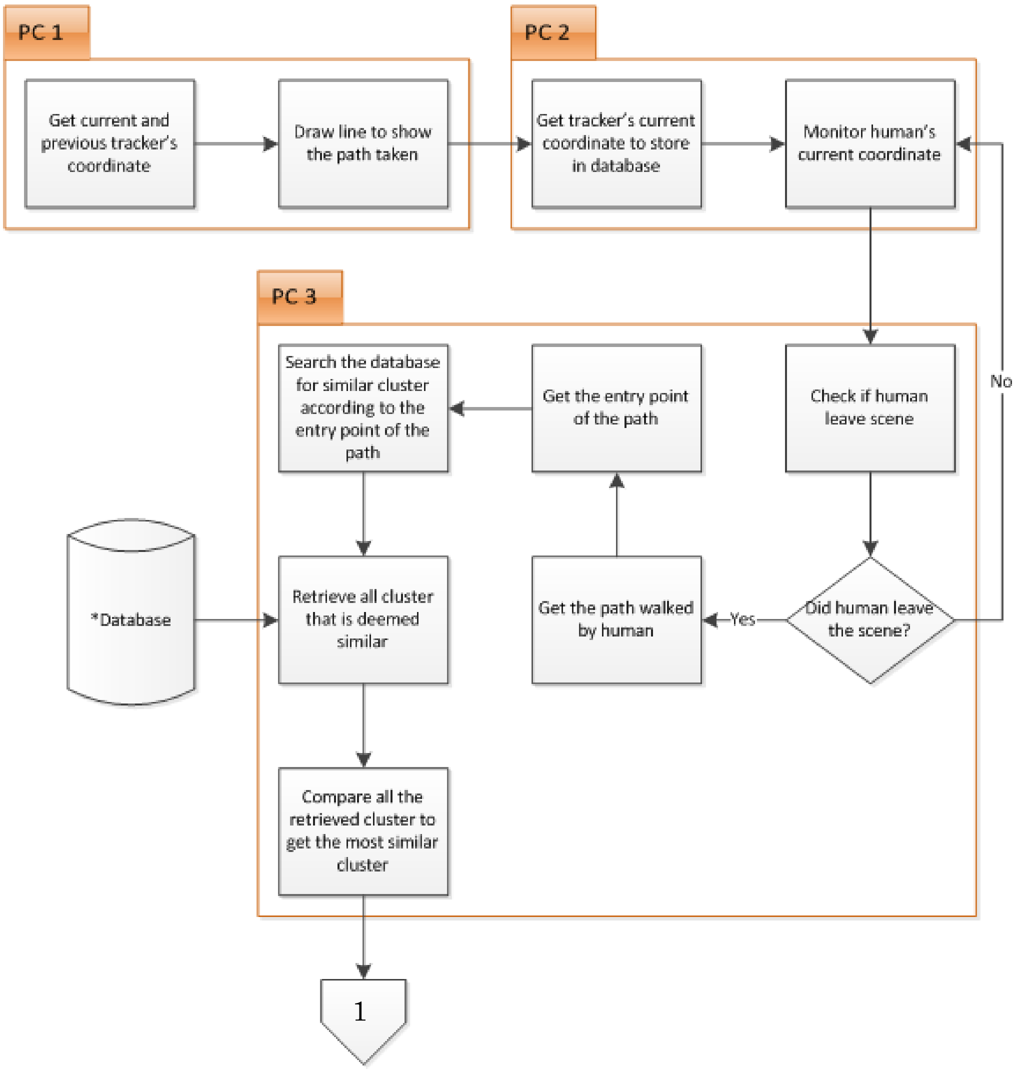
Figure 3.8: Bounding Box Image with Tracking Point (TP)

In order to obtain the TP, a simple calculation is done. The calculation is shown as follows:-

Coordinate X = Median of Width of Bounding Box

Coordinate Y = Height of Bounding Box * 0.9

Figure 3.9 and Figure 3.10 shows the breakdown of the process found in the Path Detection and Clustering Module.



PC 1
Get current and previous tracker's coordinate
Draw line to show the path taken
PC 2
Get tracker's current coordinate to store in database
Monitor human's current coordinate
PC 3
Search the database for similar cluster according to the entry point of the path
Get the entry point of the path
Check if human leave scene
No
*Database
Retrieve all cluster that is deemed similar
Get the path walked by human
Yes
Did human leave the scene?
Compare all the retrieved cluster to get the most similar cluster
1

Figure 3.9: Module 2 Structure Diagram part 1

*Database drawn in both part 1 and part 2 are the same Database

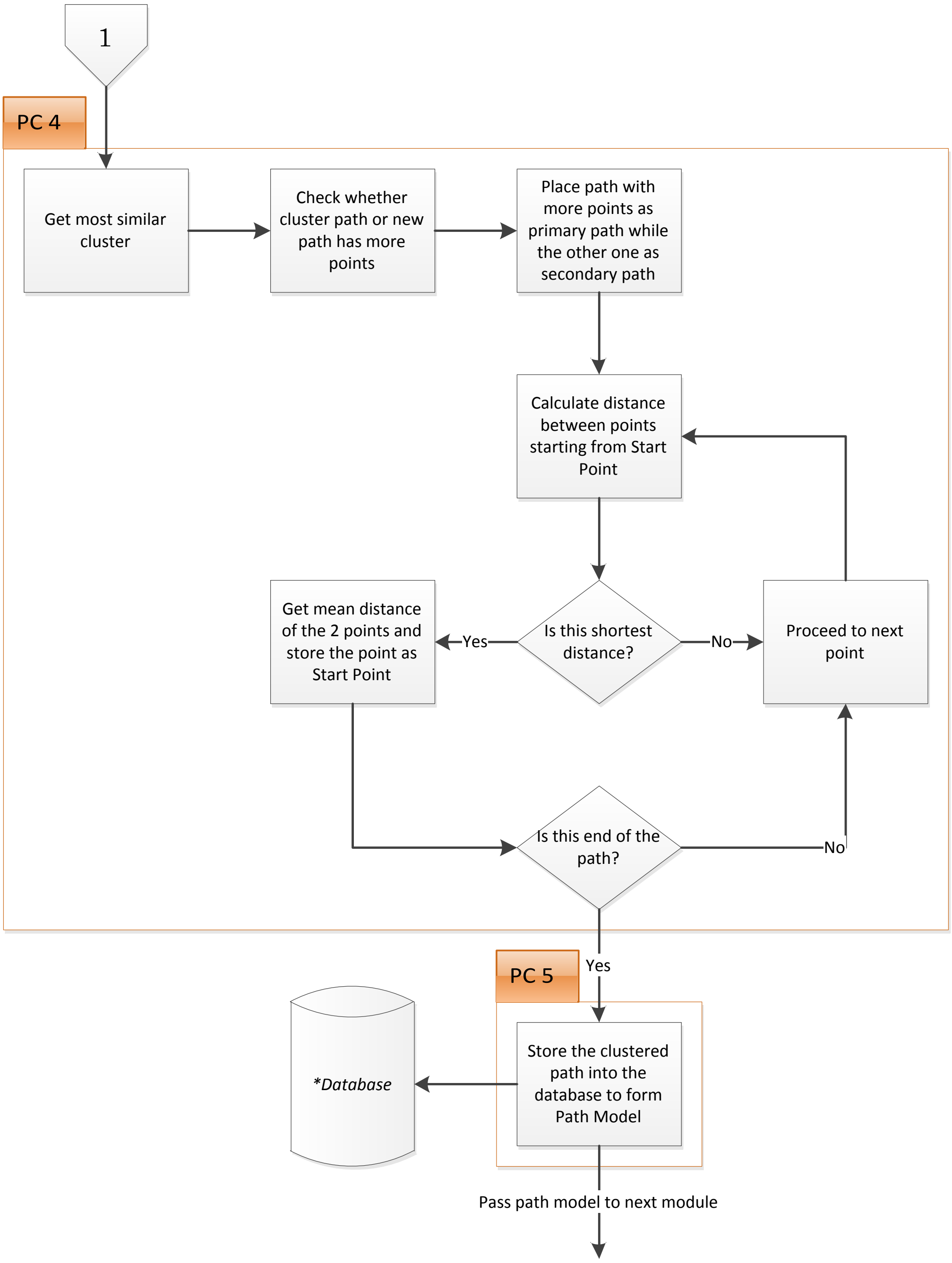

1
PC 4
Get most similar cluster
Check whether cluster path or new path has more points
Place path with more points as primary path while the other one as secondary path
Calculate distance between points starting from Start Point
Get mean distance of the 2 points and store the point as Start Point
Yes
Is this shortest distance?
No
Proceed to next point
Is this end of the path?
No
Yes
PC 5
*Database
Store the clustered path into the database to form Path Model
Pass path model to next module

Figure 3.10: Module 2 Structure Diagram part 2

*Database drawn in both part 1 and part 2 are the same Database

3.3.2.1. Draw the line to indicate what path is being taken by individual (PC 1)

After the human is detected and tracked in Module 1, all the tracker’s information will be passed on to this module for the path detection and clustering process. But before anything is being done, the path that is taken by the tracked human must first be drawn so that user can know what path exactly did the tracked human take and which cluster is the closest cluster for the newly introduced path. All of this is done for the sake of calculating the accuracy and the efficiency of the system. Firstly, after the tracker’s information is received, the current and previous coordinates of the tracker will be retrieved and a colored line will be drawn on the path that is taken by the tracked human. Figure 3.11 shows the illustrated image of the line drawn to indicate which path is being taken.

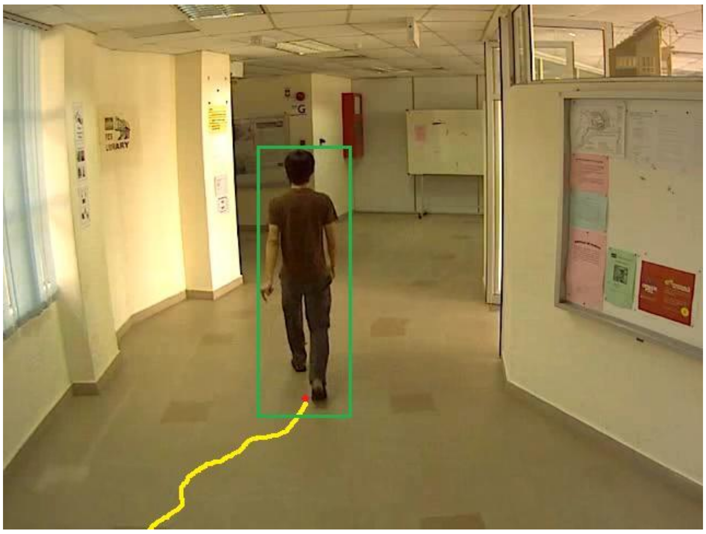


Figure 3.11: Path Taken Image

3.3.2.2. Store the coordinates of the individual into the path vector (PC 2)

By using the tracker, the exact position of the tracked human can be obtained easily. Thus, each pair of coordinates of the tracker for each frame will be stored in a database. All the coordinates that is stored are being stored as CvPoints. CvPoints are points that include both X-coordinates and Y-coordinates. Storing them separately is not only tedious but also might cause a problem in retrieving them for further analysis. Thus, the coordinates for the tracker for each frame are being stored as CvPoints.

With this coordinates, the tracked human's position can be monitored. The monitoring of the tracked human's position is required because the extraction of the path that is taken by the tracked human can only commence the moment the tracked human leaves the scene. To determine whether the tracked human has left the scene, the coordinate of the current position of the tracked human will be retrieved. If the current coordinates touches the border of the scene, then the tracked human will be deemed as left the scene. As mentioned in the first module, the size of the monitored scene's image is 320 pixels x 240 pixels. Therefore, if the X-Coordinates exceed the value of 319 or the Y-Coordinates exceed the value of 239, then the tracked human will be deemed as left the scene.

3.3.2.3. Search for cluster that is most similar to the new path after the individual leave the scene (PC 3)

The moment the tracked human is deemed to have left the scene, the path that is taken by the human will be retrieved. This is done by retrieving all the path's coordinates of the tracker. Once we have retrieved the coordinates, a comparison will be done between the freshly introduced path's entry zone (first point's coordinate) with all the available cluster's entry zones and exit zones (first point and last point's coordinate). Only clusters that have similar coordinate with the freshly introduced path will be retrieved. Finally, a comparison will be done in order to find out which remaining cluster that is not excluded is the most similar with the freshly introduced path. A path is deemed as similar if there is a pair of points; one from the clustered path and one from the freshly introduced path, where the distance between the two points does not exceed a certain threshold. If the distance exceeds the threshold, then this cluster and the path will be deemed not similar and the next cluster will be retrieved to be compared with the freshly introduced path. If after comparing with the entire cluster, not one cluster can be deemed as similar, then this freshly introduced path is a new path.

3.3.2.4. Perform path clustering (PC 4)

As soon as the system has found the cluster that is most similar with the freshly introduced path, the path clustering function will then commence by using the clustering function which employs a new and improved method of clustering path. Dynamic Time Warping (DTW) is used to cluster the paths together (E.J.Keogh and M.J.Pazzanin, 2000) and the algorithm used is derived from Levenshtein Distance algorithm. Dynamic time warping (DTW) is a method for measuring similarity between two sequences which may vary in path lengths. The reason this method is used is because it is not computationally intensive and it can cluster similar trajectory in a very short amount of time. (Zhang Zhang et al., 2006) Furthermore, when there are 2 paths with different number of points, the system will still be able to cluster the paths to form a normal path.

The basic requirement for Dynamic Time Warping is the coordinates that are obtained from both the clustered path and also the newly obtained path. We will calculate the distance of each pair of coordinates from both path and each pair of coordinates will be saved in an array and thus 2 arrays are obtained. One of the existing algorithms that are used for path clustering is known as Levenshtein Distance algorithm. The algorithm that we will be using, Enhanced Levenshtein Distance (ELD), is derived based on the existing Levenshtein Distance to compare both coordinates. Levenshtein Distance is a well-established algorithm for clustering similar path together. It is used to clustered path in order to create a Generic topographical scene description that is meant to be used as a reference to determine the status of an individual that is being tracked in the video.

Using ELD has 2 advantages. The first advantage will be both paths do not need to have the same amount of points to be compared and the second advantage is the process of clustering similar path will be made easier. By using Levenshtein Distance, although both path may not have the same amount of points, a median path can still be obtain by getting the median points of the two nearest point as shown in Figure 3.12.

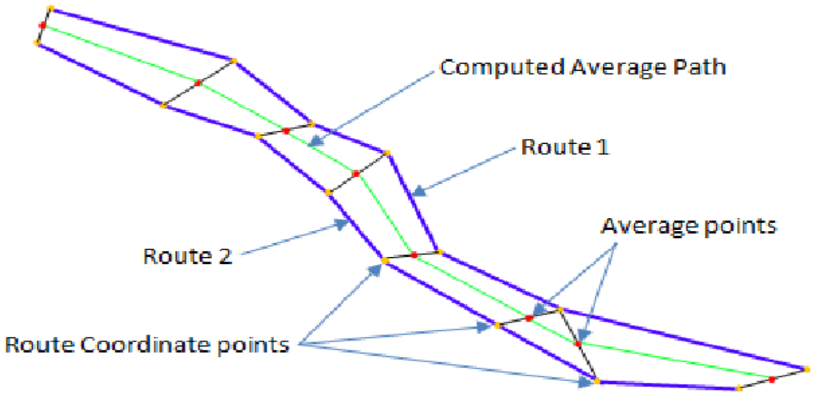


Figure 3.12: Levenshtein Distance

The output for this module is the generic topographical scene description that will eventually be used as reference guide. A normal path means that the path that is commonly and frequently taken by human. The Path Detection and Clustering Module is a training module where the purpose of the module is geared towards defining which path is the normal path in order to create a normal path reference guide known as the generic topographical scene description for the usage of the Suspicious Path Detection Module.

3.3.2.5. Store all clustered path into the database to form a Path Model (PC 5)

Once the freshly introduced path has been successfully clustered into the most similar cluster, the new clustered path will be stored into the database. At the end of this module, with all clustered paths stored in the database, a Generic topographical scene description can be formed. By inputting all the coordinates from the clustered path, a clustered path can be formed in a binary image (black and white image) where the path drawn on the image will be white in color while the background will all be in black. The Generic topographical scene description that is formed will be passed on to Module 3 for trajectory based activity analysis purposes.

The reason an image is chosen to be the Generic topographical scene description is because it will be easier for Module 3 to determine whether the tracked individual has deviated from the path or through a simple comparison of the coordinates of the Generic topographical scene description’s normal path and the coordinates of the path taken by the individual tracked in the scene. The Generic topographical scene description will ultimately act as a reference guide for Module 3 to determine which path is not normally taken. Without this Generic topographical scene description, the tracked human that takes the path that is not commonly taken may go undetected.

Figure 5.2 will show an example of the image of a Generic topographical scene description that is produced when all the similar paths have been grouped together in their respective clusters. In this example, there are currently 3 clusters in the Generic topographical scene description and the Generic topographical scene description will then be passed on to the next module to act as their reference guide to determine which path is a normal path.

3.3.3 Module 3 Suspicious Path Detection

This module is mainly on the detection of individual's gait that is currently being monitored. The breakdown of the Suspicious Path Detection Module is shown in Figure 3.13.

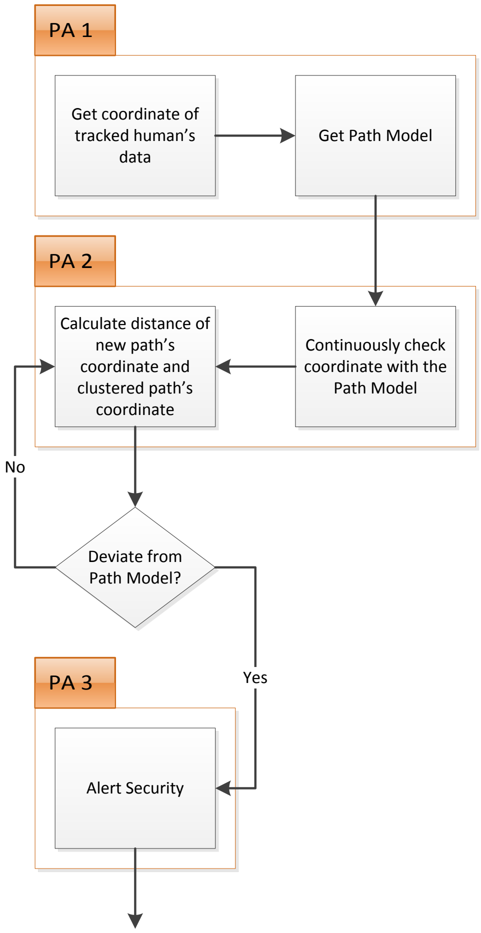

PA 1
Get coordinate of tracked human's data
Get Path Model
PA 2
Calculate distance of new path's coordinate and clustered path's coordinate
Continuously check coordinate with the Path Model
No
Deviate from Path Model?
Yes
PA 3
Alert Security

Figure 3.13: Module 3 Structure Diagram

Figure 3.13 shows the detailed breakdown of the processes that occur in the Suspicious Path Detection Module. Module 3 will only be executed during the testing process. This module’s main purpose is to analyze the path that is taken by the tracked human and alert security personal should there be any tracked human taking the path that is not commonly taken. The moment Module 3 is executed; the Generic topographical scene description that is trained through Module 2 will be loaded into the system for reference in order to determine whether the path that is taken by the tracked individual in the scene is normal or deviated from the Generic topographical scene description.

3.3.3.1. Obtain information from Human Detection Module (PA 1)

The moment a new moving object is detected in the scene through background subtraction and contour detection, the human detection function in Module 1 (HD 2) will be used in order to determine whether the moving object that is detected in the scene is human or not. If the moving object is deemed to be human by the human detection function in Module 1 (HD 2), then this confirmation will then be passed on to the human tracking function.

Once the moving object has been deemed to be human, that human will be set a track point and will be continuously monitored. The system will retrieve the current coordinate of the human by retrieving the tracker’s information that is obtained through Module 1 (HD 2) in order to know where the tracked human is located in the scene. In order to draw a box as stated in Module 1 (HD 3) a set of coordinates must be obtained and these coordinates are used to continuously track the human in the scene.

3.3.3.2. Continuously check the current coordinate of the Tracking Point (PA 2)

With the coordinates obtained, the current position of the tracked human will be constantly compared with the Generic topographical scene description's coordinate to determine whether the tracked human is taking a path that is not commonly taken or not. If the coordinate of the current position of the tracked human falls on the white region in the Generic topographical scene description's image, then the tracked human is still walking on the normal path as determined through training in Module 2. If the coordinate of the current position of the tracked human falls on the black region in the Generic topographical scene description's image, the tracked human will be deemed as taking a path that is not commonly taken.

3.3.3.3. Alert Security (PA 3)

Once the tracked human has been determined of taking the path that is not commonly taken, the system will them alert the security personal or guard on duty and the detected human that trigger the security alert will be monitored for further analysis by the security guard. If the human is deemed as potentially posing security risk, further security means of action will be taken.

3.4 Conclusion

The system’s internal function and all the methods used in this system are explained in details and the overall system architecture is introduced with the flow of the complete system shown. Figures and tables are also used to help understand the system’s internal working and the program’s flow. A general explanation for each module is also given. In conclusion, the system that is being handled focuses on the detection of tracked human taking unusual path by using the Generic topographical scene description that is formed.

CHAPTER 4

IMPLEMENTED SYSTEM ENHANCEMENT

4.1 Introduction

This chapter will discuss about the contribution of the system. A detailed discussion about the Enhanced Levenshtein Distance (ELD) and the similar cluster searching index method is included in this chapter. This chapter will show how the ELD and the similar cluster searching index method work in this system by using tables and figures.

4.2 Enhanced Levenshtein Distance (ELD)

Dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences which may vary. The basic problem that DTW attempts to solve is how to align two sequences (e.g. similar path) in order to generate the most representative distance measure of their overall difference (e.g. clustered path). If you have two paths encoded as a sequence of evenly spaced values (representing, for example, the coordinates of the tracked individual of each frame), then an obvious way to compare the signals is to sum the differences in coordinate at each point along the paths. However, a problem arises if there is any discrepancy in the alignment of the paths – if for example one of the paths is longer or shorter compared to the other.

An example of DTW clustering method is Levenshtein Distance method. A many-to-many matching between the points in path x and the points in path y matches every point x_i in x with at least one point y_j in y, and every point in y with at least a data point in x. The set of matches (i, j) forms a warping path ɯ. The Levenshtein Distance is defined as the minimization of the l_p norm of the differences $\{x_i - y_j\}_{(i,j) \text{in } ɯ}$ over all warping paths. A warping path is minimal if there is no subset ɯ' of ɯ forming a warping path. For path clustering, it is required that all warping paths to must be minimal.

In computing the Levenshtein Distance, it is commonly require the warping to remain local. For path x and path y, values x_i and y_j is aligned only if $|i - j| < w$ for some locality constraint $w \geq 0$. When $w = 0$, the Levenshtein Distance becomes the l_p distance whereas when $w \geq n$, the Levenshtein Distance has no locality constraint. The value of the Levenshtein Distance diminishes monotonically as w increases. By dynamic programming, the Levenshtein Distance requires $O(N^2)$ time. To compute the Levenshtein Distance, we use the recursive Euclidean Distance formula. The Euclidean Distance between point p and q is the length of the line segment PQ where point p is the current coordinate while point q is the previous coordinate. In Cartesian coordinates, if $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ are two points in Euclidean n-space, then the distance from p to q is given by:-

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \text{ ……………..….. (2)}$$

Not only is this method computationally intensive, it will also slow down the clustering function if there are a large number of points to cluster. Thus, the new method that is going to be used is an improved version of the Levenshtein Distance that will reduce the number of computations required. Firstly, before the path clustering function starts, the freshly introduced path and the clustered path will be compared to see which path has the larger number of points in it. The path with the larger number of points will be deemed as the primary path while the other path will be deemed as a secondary path. The pseudo-code for this function is shown below:-

```
int     SimilarCluster(InputPath:     vector<CvPoint>     [0..n],
ClusterPath:  vector<vector<CvPoint>>  [0..m],  memoryindexpath:
vector<int> [0..p]){

      FOR i = 0 to m

            IF inputpath > clusterpath
            Compute DTW with inputpath as primary path
            ELSE
            Compute DTW with clusterpath as primary path
            ENDIF

            FOR j = 0 to q

                  IF inputpath > selected clusterpath
      Compute distance with inputpath as primary path
                  ELSE
      Compute DTW with selected clusterpath as primary path
                  ENDIF

                  IF cost > threshold
                        Skip to next path
                  ENDIF

                  FOR all the value in the vector
                  Store the value into the clusterindex
                  ENDFOR

            ENDFOR

      ENDFOR

      RETURN clusterindex
}
```

If the primary path happens to have fewer points than the secondary path, the final outcome of the path clustering function will be a clustered path with a lesser number of points. The system must make sure that the number of points in the clustered path will only increase and not decrease so that the accuracy for the Generic topographical scene description will not decrease. Now that the path with more points and the path with fewer points have been determined, the path clustering function can now begin.

The distance between the points in both primary path and secondary path must be calculated. The pseudo-code for this function is shown below:-

```
vector<int> ComputeDTW(Path1: vector<CvPoint> [0..n], Path2:
vector<CvPoint> [0..m]){

      FOR i = 0 to m
            Insert value 0 into the temp vector

      FOR i = 0 to n
            Insert value into the DTW vector

            IF this is the first loop
                  Begin from the beginning of the vector
            ELSE
                  Begin from the previous loop’s last position
            ENDIF
            FOR j = startpoint to m

                  Compute distance between the 2 paths
                  Store the distance into the DTW vector

                  IF the distance is less than the minimumvalue
                        The distance will be the minimumvalue
                  ENDIF

            IF path has been fully tranversed
                  Store the value into the memoryindex vector
            ENDIF

            ENDFOR

      ENDFOR

      ENDFOR
      RETURN memoryindex;
}
```

```
vector   <CvPoint>   ClusteringPath(InputPath:   vector<CvPoint>
[0..n],  ClusterPath:  vector<CvPoint>  [0..m],  memoryindexpath:
vector< int > [0..p]){

      FOR i = 0 to p
            Store all the vector value into the memoryindex

      FOR i = 0 to q

            IF inputpath is longer than clusterpath

            Get midpoint of both path with inputpath as primary
path

            ELSE

            Get  midpoint  of  both  path  with  clusterpath  as
primary path

            ENDIF

      ENDFOR

      ENDFOR

      RETURN newpath
}
```

The distance will be computed using the Euclidean Distance. The method to calculate the distance is shown in Equation 2. For Levenshtein Distance, normally all the points in both paths will be traversed in order to search for the shortest distance between the 2 paths for each pair of points. However, for the new and improved method, not all points in the secondary path need to be transverse. Firstly, the first point in the primary path will transverse all the points in the secondary path in order to find the points that are closest to it. Once the point in the secondary path is found, the calculation will proceed to the next point in the primary path. However, unlike the original Levenshtein Distance's method, the calculation for the distance will not begin from the first point of the secondary path. The starting point will be at the point where the previous primary path's point found to be the closest point. Thus, after repeating the process of calculation, the more points in the primary path that have been calculated, the fewer points will be needed to be transverse in the secondary path.

By reducing the full computation of the Levenshtein Distance, at most (N^2 - R) (where R is the number of ignored computations) comparisons between points are required to cluster the path containing N points. By using this method, the number of calculations that needs to be done will be lesser when compared with the original Levenshtein Distance method. The table below shows how the calculation works for clustering two different paths with different number of points in it:-

Table 4.1: Calculation of Dynamic Time Warping

	(X_1,Y_1)	…	(X_{1+a},Y_{1+a})	…	(X_{1+b},Y_{1+b})	…	(X_m,Y_m)
(X_1,Y_1)	O	…		…		…	
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
(X_{1+c},Y_{1+c})	X	…	O	…		…	
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
(X_n,Y_n)	X	…	X	…	X	…	O

The “X” means that the particular points can be ignored and no calculation needs to be done. The “O” means that that particular point is the best point or the closest point and a mean distance needs to be calculated so that a new cluster can be formed. By using the ELD, the more the points are transverse, the lesser the number of computation is required.

There is a similar method that also applies the limitation of computation called the Sakoe-Chiba band. (Sakoe Chiba, 1978). Sakoe-Chiba band applies the global path constraints when comparing time series where i and j in w_m = (i; j) are constrained to ($j - r \leq i \leq j + r$), where r is the value of the path constraint that is used. Thus, two time series U and L (for upper and lower) can be defined and an envelope that the time series must lie in can be created.

The envelope that is created has a constant distant between the upper and the lower bound. When this is being applied to path clustering, it is also capable of reducing the amount of computation. However, in terms of efficiency, the ELD is a better choice since it does not apply a constant distance envelope but a dynamic lower bound. While Sakoe-Chiba band applies the envelope where i and j in w_m = (i; j) are constrained to ($j - r \leq i \leq j + r$) where r increases geometrically, the ELD applies a similar envelope but with a dynamic lower bound where i and j in w_m = (i; j) are constrained to ($j - k \leq i \leq j$) where k will increase exponentially the more it transverse the paths. The more it transverse through the path, the lesser the number of computation it is required to do.

4.3 Similar Cluster Searching Index

In order to further reduce the number of computations, a similar cluster searching index function is applied. In order to search for the most similar path to cluster with, if there is N number of path where Cluster Path = {$p_i[x_i,y_i]$,…, $p_n[x_n,y_n]$ available to be compared with, all N paths must be included into the computation. However, if a selective process is done on the path available and only the selected paths are retrieved and compared with while the remaining paths are all discarded, then the number of computations required to cluster the path is reduced. Without the cluster searching index function, in order to search and cluster a path with a generic topographical scene description that has M paths requires $O(MN^2)$ time where N is the number of points in the path and M is the number of path that is in the generic topographical scene description. However, by applying the cluster searching index function, only $O((M-K)N^2)$ is required where K is the number of ignored path in the generic topographical scene description.

4.4 Conclusion

The contribution and the methods that are used in this system are explained in details. Formulas and tables are also used to help understand the contribution and the process flow of the method. In conclusion, the system will be implementing a new and more efficient way of path clustering in order to better form a Generic topographical scene description.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1. Introduction

This chapter will discuss the evaluation of the proposed system. A detailed discussion about the outcome of the proposed system and the analysis of the result obtained is also included in this chapter. The self captured videos that are used as the testing data to test the proposed system is captured in 3 different scenes with 3 different scenarios. The first scenario will be captured with 1-2 humans appearing in the scene at the same time. The second scenario will be captured in a different scene with similar condition; 1-2 humans appearing at the same time in the scene. The third scenario will be captured in another scene but this time with 5-6 humans appearing in the scene at the same time.

5.2. Testing Environment

The proposed system's target is for the implementation in the real-time environment where scenes captured are being processed and analysed in real-time. Therefore, the live testing of the proposed system is carried out in a university compound area. Equipment (i.e. camera, laptop, etc) that are necessary and used for the live testing and tools (i.e. Visual Studio, OpenCV, etc) are set up in the testing ground. The purpose and use of the equipment and tools will be explained in detail in the following subsection.

5.2.1. Testing Equipment

The equipment that is used in the testing process includes a computer that is installed with the proposed system and is connected to a camera. The computer will be used to process and analyse the captured scene. In order to reduce the noise and possibility of data lose, it is preferable to set up the computer near the camera as the camera is connected to the computer via a LAN Cable.

The camera will be used to capture the testing ground's activity. The camera needs to be placed in a high location using a high stand and will be placed facing the ground at a rough estimation of 30° to 35° of inclination in order to best reproduce the image that is captured by the Closed-Circuit Television (CCTV). The camera will also be static and not moving. This is to obtain a better background image quality as stated in Chapter 3. The configuration of the equipment is shown in Figure 5.1.

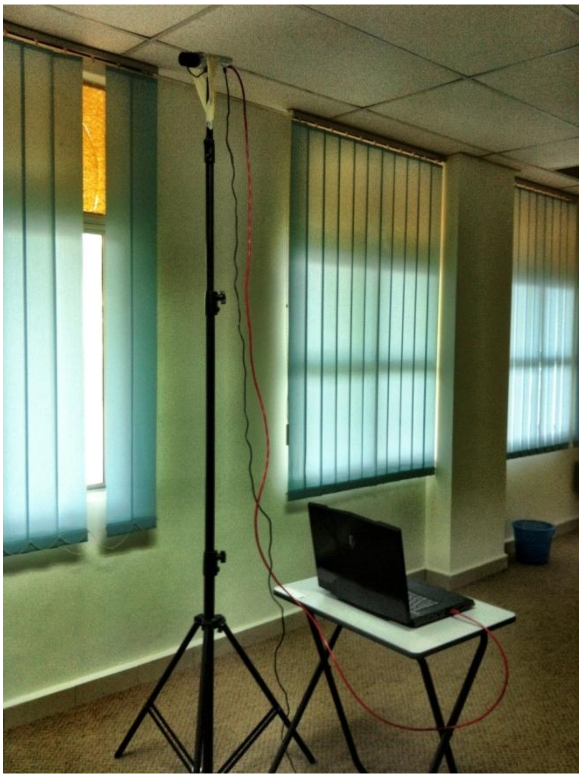


Figure 5.1: Equipment Setup

5.2.2. Testing Tools

The tools that are used in the testing process include the main tool (i.e. Visual Studio Professional Edition 2008) and also two libraries (i.e. OpenCV 2.0 library and SVMLight library) that are being used in order for the proposed system to function properly. The OpenCV 2.0 library are used mainly for the purpose of processing the images that are acquired while the SVMLight library are used for the sole purpose of building the SVM classifier that is used to detect human that may appear in the scene.

5.2.3. Path Model Image

The generic topographical scene description that is created through the training process as mentioned in Chapter 3 will be loaded into the proposed system. The image will be in black and white with an image size of 320 pixels (Width) and 240 pixels (Height). The clustered path that indicates the path that is normally taken by human will be white in colour while the rest of the background will be black in colour. Each time the proposed system runs the training process, the generic topographical scene description will be updated. An example of the generic topographical scene description is shown in Figure 5.2.

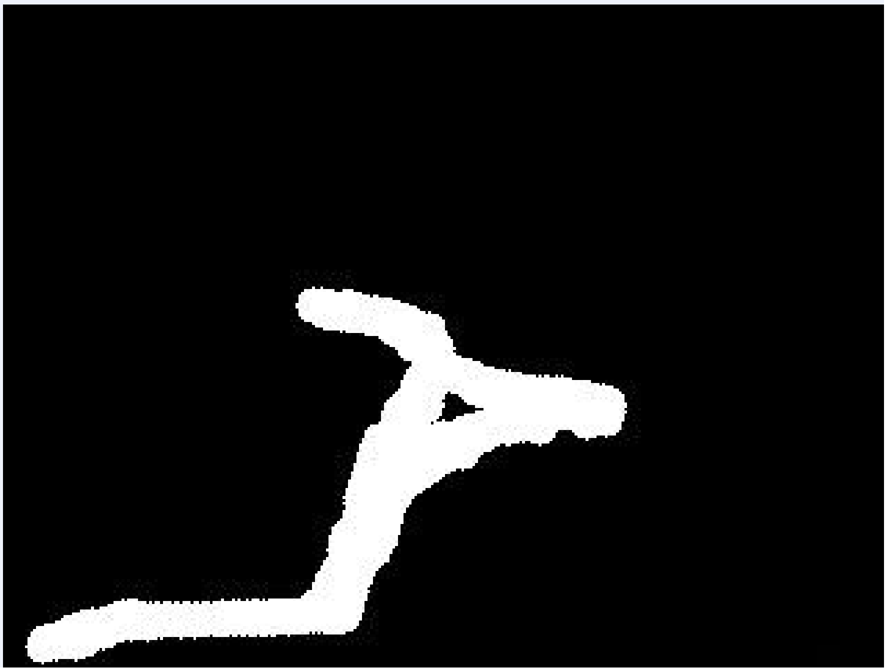


Figure 5.2: Path Model

5.2.4. Test Data

For the sole purpose of testing the proposed system, all the data that is used are self-captured. 21 videos were used as test data where 9 of the videos are used to test the Path Clustering efficiency (number of computations) and accuracy while the remaining 12 videos were used to test the Trajectory Based Activity Analysis. The test data consist of people that perform a certain actions that are predetermined based on the module of the proposed system, i.e. videos of people walking in and out of the scene using different paths or videos of people going through and back using the same path. The preparation and the detailed description of the test data will be discussed in the following subsection.

5.2.4.1. Test Data Description

The setup at the university ground during the testing process is shown in Figure 5.1. For the test data capturing, the number of humans in the scene that appear in the same time will be between 1-6 people. Furthermore, the lighting condition of the scene must be regulated or controlled. Finally, the entry and exit zone are indicated by the black rectangles and labelled A, B, C and D as shown in Figure 5.3, Figure 5.4 and Figure 5.5.

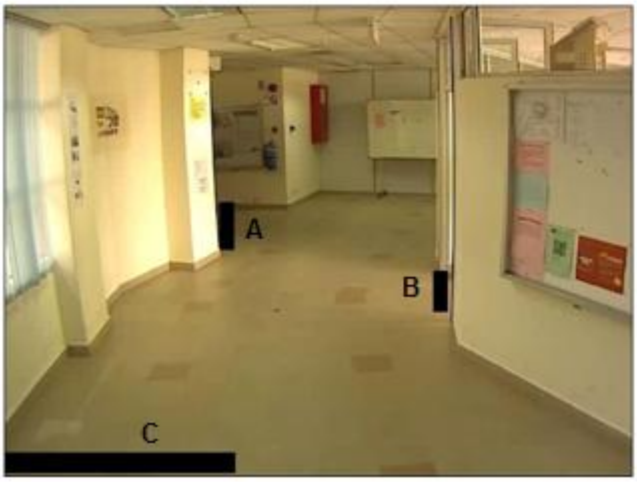


Figure 5.3: Entry and Exit zones Scene 1

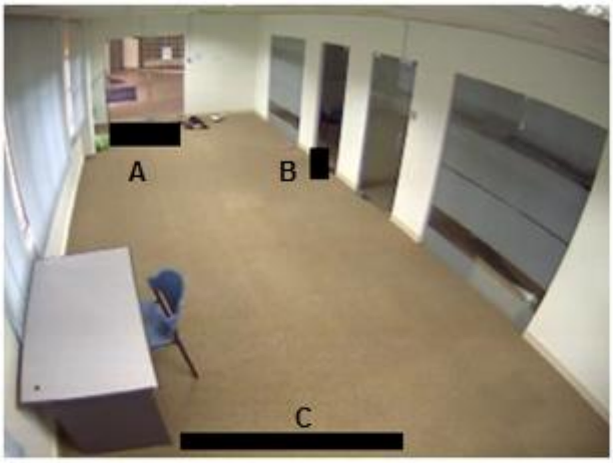


Figure 5.4: Entry and Exit zones Scene 2

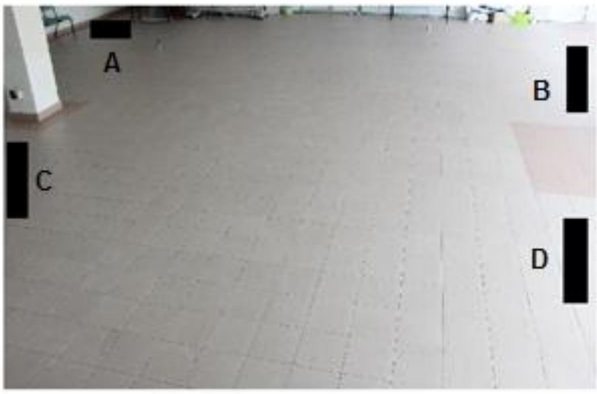


Figure 5.5: Entry and Exit zones Scene 3

5.3. System Evaluation Based On Test Cases

The evaluation process will be divided into 3 processes. Each process will evaluate different criteria. The results and discussion of each criterion will be provided in each subsection respectively. The test data that will be used in each criterion are of same resolution, i.e. 320×240 pixels and frame per second, i.e. 30. However, each video varies in length.

5.3.1. Number of computations

In this section, the evaluation of the proposed system will be based on the path clustering algorithm called Dynamic Time Warping. As mentioned in Chapter 3, Dynamic Time Warping (DTW) is a well-established algorithm for clustering similar path together (E.J.Keogh and M.J.Pazzanin, 2000) and the name of the method that is going to be used to cluster similar path based on Dynamic Time Warping algorithm is a new method of clustering that is based on Levenshtein Distance called the Enhanced Levenshtein Distance. For this section, we will be using 9 videos as data sets that are taken from 3 different scenes with the description of each video shown in Table 5.1.

Table 5.1: Video Data Set Properties

Attribute		Video Format	Dimension	Frames Count	Video Length
Scene 1	Test Case 1	*Avi*	320 x 240	1575	00:00:52
	Test Case 2	*Avi*	320 x 240	1117	00:00:37
	Test Case 3	*Avi*	320 x 240	1117	00:00:37
Scene 2	Test Case 1	*Avi*	320 x 240	1297	00:00:43
	Test Case 2	*Avi*	320 x 240	1580	00:00:52
	Test Case 3	*Avi*	320 x 240	1772	00:00:59
Scene 3	Test Case 1	*Avi*	320 x 240	2043	00:01:08
	Test Case 2	*Avi*	320 x 240	1832	00:01:01
	Test Case 3	*Avi*	320 x 240	1922	00:01:04

The evaluation will be done in two methods. This section will be doing the comparison of the number of computations needed to cluster the path between the Levenshtein Distance method and the Enhanced Levenshtein Distance that is used in the proposed system while the next section will be calculating the warping accuracy of the method based on a paper written by Zhang Zhang, Kaiqi Huang and Tieniu Tan. (Zhang Zhang et al., 2006)

There will be 3 entry and exit zones for scene 1 and 2 but 4 entry and exit zones for scene 3 as shown in Figure 5.3, Figure 5.4 and Figure 5.5. For path clustering, a path is said to be similar if and only if two of their nearest points of the two paths does not exceed a certain distance and they have the same entry and exit zone. A comparison between the original Levenshtein Distance method with the Enhanced Levenshtein Distance method will be shown in the table below. In order to calculate the percentage that is reduced, the formula that is used is shown below:-

$$Percentage\ Reduced = \frac{Original\ \ Total - Remaining\ \ Total}{Original\ \ Total} \times 100\% \ \ldots\ldots\ldots\ldots\ldots\ldots..\ (3)$$

The percentage reduced indicates the percentage of number of computations that has been reduced while the original total and the remaining total indicates the original total of computation required to cluster the path and the new total of computation required by using the new method to cluster the path respectively.

Table 5.2: Computation Results

Scene	Cluster	Total Computation	Needed Computation	%	Reduced Computation	%
1	1	47064	23899	50.78	23165	49.22
	2	67860	36473	53.75	31387	46.25
	3	28224	16688	59.13	11536	40.87
	4	23712	12263	51.72	11449	48.28
	5	24649	12245	49.68	12404	50.32
	6	20449	9854	48.19	10595	51.81
2	1	19881	11693	58.81	8188	41.19
	2	20736	9692	46.74	11044	53.26
	3	28900	15733	54.44	13167	45.56
	4	25921	13788	53.19	12133	46.81
	5	62500	36241	57.99	26259	42.01
	6	72900	41397	56.79	31503	43.21
3	1	94240	42199	44.78	52041	55.22
	2	92412	49432	53.49	42980	46.51
	3	95477	47023	49.25	48454	50.75
	4	90600	44761	49.41	45839	50.59
	5	103680	52547	50.68	51133	49.32
	6	85550	41056	47.99	44494	52.01
	7	85260	42074	49.35	43186	50.65
	8	84000	44213	52.63	39787	47.37
	9	95790	48975	51.13	46815	48.87
	10	101757	48396	47.56	53361	52.44
	11	87314	41056	47.02	46258	52.98
	12	95784	46009	48.03	49775	51.97

As shown in the Table 5.2, it is shown clearly that by implementing the Enhanced Levenshtein Distance method, the number of computations will be reduced between 40.87% and 55.22%. The number of computations that is required has been reduced almost by half from the original number of computations needed by using the original Levenshtein Distance method.

5.3.2. Clustering Accuracy

Trajectory clusters are often represented by routes or paths most commonly taken by motion objects. (Z.Fu et al., 2005) According to the principle, we chose the scene shown in the Figure 5.6, Figure 5.7 and Figure 5.8.

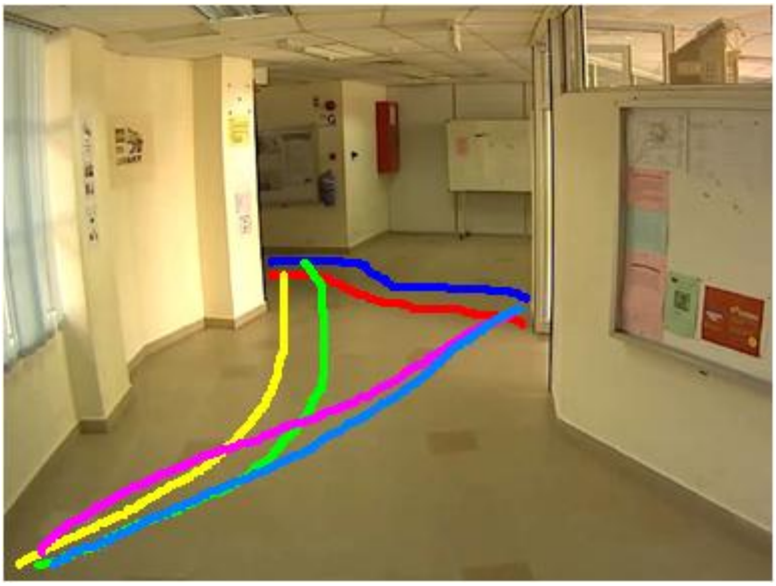


Figure 5.6: Ground Truth for Scene 1

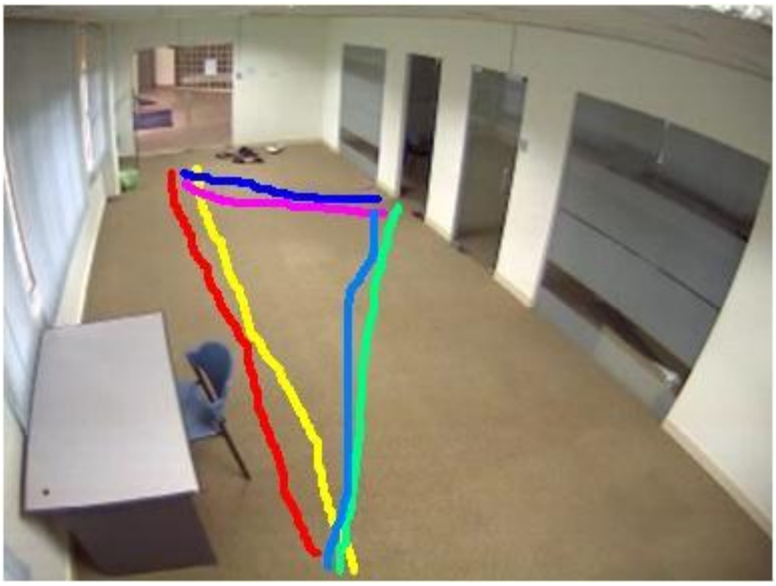


Figure 5.7: Ground Truth for Scene 2

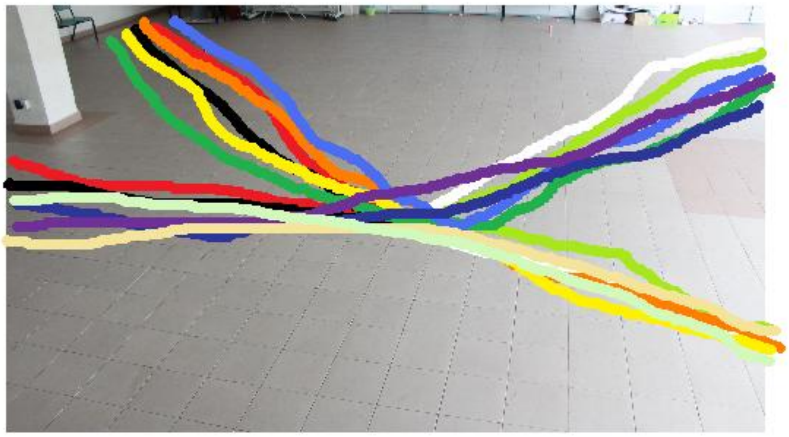


Figure 5.8: Ground Truth for Scene 3

This calculation is done in order to see how accurate the clustering algorithm is. (Zhang Zhang et al., 2006) All the normal paths from the 3 different scenes are labelled manually. In scene 1, the shortest length of the cluster is 143 points while the longest length of the cluster is 261 points. In scene 2, the shortest length of the cluster is 146 points while the longest length of the cluster is 273 points. In scene 3, the shortest length of the cluster is 294 points while the longest length of the cluster is 329 points. Figure 5.6, Figure

5.7 and Figure 5.8 illustrate the colour lines in scene 1, scene 2 and scene 3 where each line denotes a cluster.

Let G be the ground truth set with N clusters, C be the resulting clusters set. Given a cluster c_i that exists in C, the corresponding cluster g_m that exists in G that maximizes $|C_i \cap g_j|$ can be found. Then Correct Cluster Rate (CCR) is defined to evaluate the accuracy of each cluster by using the formula shown below:-

$$CCR = \frac{1}{N} \sum_{i}^{N} P_i \text{ ………….. (4)}$$

N is the total number of clusters formed, and P_i denote the accuracy of the correct clustered trajectories in *i*-th resulting cluster. The accuracy is computed by calculating the deviancy rate between the trained clusters and the cluster obtained from the ground truth. Deviancy Rate is the difference distant between the trained cluster and the expected cluster (ground truth). The P_i and the Deviancy Rate is computed as follows:-

$$P_i = \begin{cases} 0 \; if \; \exists c_k \in C, |c_k \cap g_m| > |c_i \cap g_m| \\ \qquad |c_i \cap g_m| \; otherwise \end{cases}$$

$$Deviancy\ Rate = \frac{1}{M} \sum_{i}^{M} Q_i \text{ ………………….. (5)}$$

M is the total number of points in the cluster and Q_i the distance difference between the calculated cluster points in the *i*-th cluster and the expected points in the *i*-th from the ground truth.

Thus, by using the formula stated above, we can calculate the accuracy of the path clustering by comparing the deviancy rate between the clusters that are formed through the training stage and the clusters that are labelled manually (ground truth). After the calculation, the results would be the accuracy of the path clustering.

The results from the CCR will be shown in Table 5.3 where G size is the length of the *i*-th cluster in the ground truth and C size is the length of the *i*-th cluster in the trained cluster. The average of all the CCR in a scene is the Clustering Accuracy (CA).

Table 5.3: Path Clustering Accuracy Results

Attribute		G size	C size	Deviancy Rate (%)	CCR (%)	CA (%)
Scene 1	Cluster 1	228	218	0.88	99.12	99.03
	Cluster 2	261	251	0.79	99.21	
	Cluster 3	168	143	1.01	98.99	
	Cluster 4	156	152	1.14	98.86	
	Cluster 5	157	157	1.01	98.99	
	Cluster 6	143	141	1.01	98.99	
Scene 2	Cluster 1	151	141	0.98	99.02	98.925
	Cluster 2	146	144	1.09	98.91	
	Cluster 3	185	170	1.11	98.89	
	Cluster 4	161	161	1.07	98.93	
	Cluster 5	260	250	1.05	98.95	
	Cluster 6	273	270	1.15	98.85	
Scene 3	Cluster 1	312	310	0.91	99.09	99.06
	Cluster 2	306	306	0.79	99.21	
	Cluster 3	318	311	0.88	99.12	
	Cluster 4	319	302	1.05	98.95	

	Cluster 5	329	324	1.01	98.99	
	Cluster 6	298	295	1.13	98.87	
	Cluster 7	294	294	0.87	99.13	
	Cluster 8	305	300	0.85	99.15	
	Cluster 9	313	310	0.93	99.07	
	Cluster 10	321	321	0.98	99.02	
	Cluster 11	299	298	1.01	98.99	
	Cluster 12	314	312	0.93	99.07	

From the results shown in Table 5.3, it is shown that the trained path is quite accurate with an accuracy of 99.03% for scene 1, 98.925% for scene 2 and 99.06% for scene 3.

5.3.3. System Evaluation for Trajectory Based Activity Analysis

In this section, the evaluation of the proposed system will be based on the trajectory based activity analysis of the path that is taken by the tracked human. The Generic topographical scene description that is created during the training process as shown in Figure 5.2 will be used as a reference to determine whether the path that the tracked human is taking is a commonly taken path or not. Since the Generic topographical scene description image's size is 320 pixel X 240 pixel which happens to be the same size as the test data's dimension, a simple process of comparison between the Generic topographical scene description's image's coordinate with the data set's coordinate will help determine which path the tracked human is taking.

A total of 12 videos with 3 scenarios that are taken from 3 different scenes will be used for the testing purpose for this section. 4 of the videos (Video from scene 1 and 2) will be showing a tracked human in the scene that is taking a path that is not commonly used while the other 4 video (Video from scene 1 and 2) will be showing a tracked human in the scene that is taking a path that is commonly taken. Another 4 videos (Video from scene 3) will show a mixture of people (some taking deviant path while others are taking normal path) taking different path at the same time. The results of the output from the 12 videos will be shown in a graph. The graph that has been prepared will show the walking pattern of the tracked human that appeared in the scene.

A "1" will be shown in order to indicate the tracked human is taking a path that is not commonly used while a "0" will indicate that the path that the tracked human is taking is commonly used. Whether the human is taking a deviant path or not is determined by the deviancy counter. A human is deemed to be taking a deviant path if the deviancy counter exceeded the threshold value.

Deviancy counter is the value given by the number of frames the human is detected taking the deviant path and the threshold value that has been set in order to determine whether the human is taking the deviant path is 10 frames consecutively. Therefore, if a human is detected taking a deviant path for 10 frames consecutively, then the human is deemed to be taking a deviant path.

The X-axis indicates the frame number (position of the frame in the video i.e. Frame no. 324...) while the Y-axis indicates the path that is taken by the tracked human ("1" for the path that is not commonly used and "0" for the path that is commonly used). The description of the 12 videos that are used in the testing is shown in Table 5.4.

Table 5.4: Data Set Properties

Attribute		Video Format	Dimension	No. of Subject	Distortion of results
Scene 1	Test Case 1	*Avi*	$\mathbf{320 \times 240}$	1	No
	Test Case 2	*Avi*	$\mathbf{320 \times 240}$	1	Yes
	Test Case 3	*Avi*	$\mathbf{320 \times 240}$	1	No
	Test Case 4	*Avi*	$\mathbf{320 \times 240}$	2	Yes
Scene 2	Test Case 1	*Avi*	$\mathbf{320 \times 240}$	2	No
	Test Case 2	*Avi*	$\mathbf{320 \times 240}$	2	Yes
	Test Case 3	*Avi*	$\mathbf{320 \times 240}$	2	No
	Test Case 4	*Avi*	$\mathbf{320 \times 240}$	2	No
Scene 3	Test Case 1	*Avi*	$\mathbf{320 \times 240}$	5	Yes
	Test Case 2	*Avi*	$\mathbf{320 \times 240}$	6	Yes
	Test Case 3	*Avi*	$\mathbf{320 \times 240}$	6	No
	Test Case 4	*Avi*	$\mathbf{320 \times 240}$	5	Yes

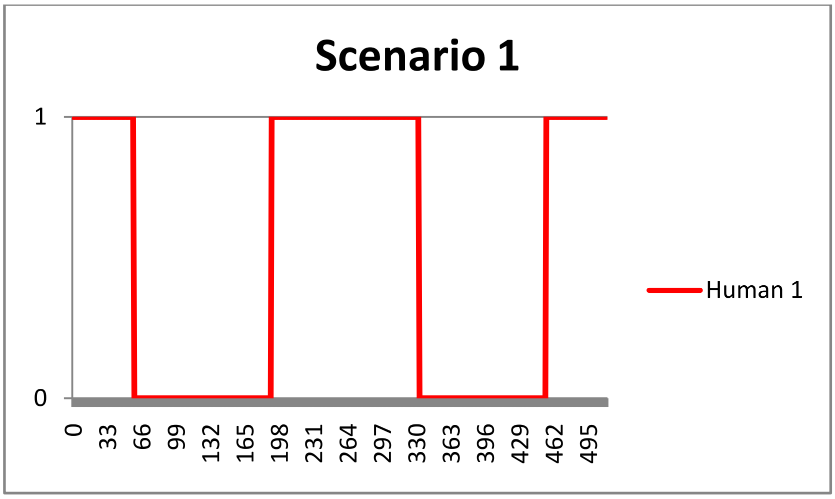


Figure 5.9: Graphical Result 1 Representative for Scene 1

In the beginning, the graph show that the tracked human is currently on the path that is taken is not commonly used. However, starting from a frame 34, when the coordinates are being compared with the Generic topographical scene description's coordinate, the tracked human's coordinate coincidentally falls within the white region in the Generic topographical scene description (white region indicates the path that is normally taken). The deviancy counter for this video has exceeded 10 frames from the beginning of the video and thus, as shown in

Figure 5.**9**, this clearly indicates that the path that the tracked human is taking is a path that is not commonly taken. So this video is clearly showing a tracked human taking a deviant path. The figure below shows the deviant path that is taken by the detected human.

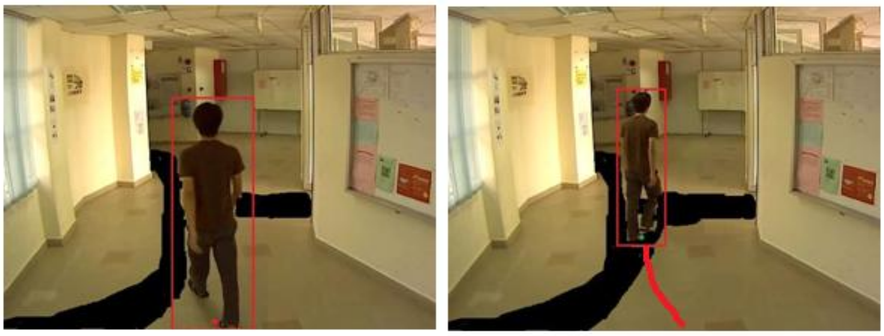


Figure 5.10: Path Taken 1 Scene 1

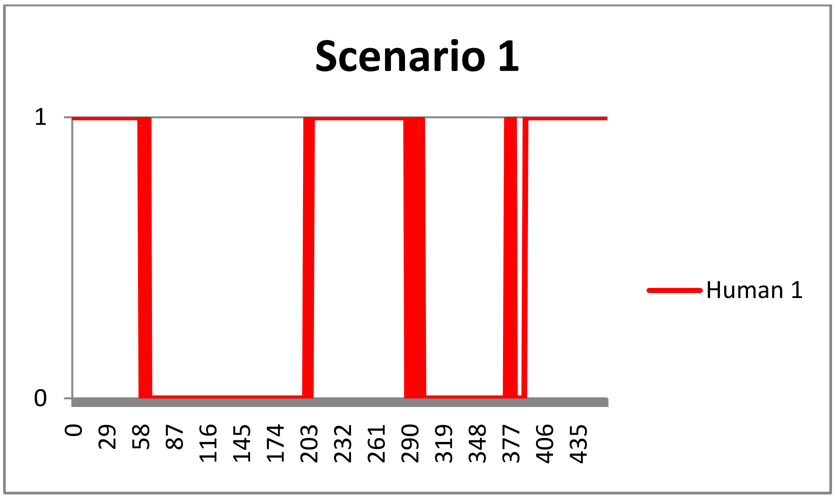


Figure 5.11: Graphical Result 2 Representative for Scene 1

In

Figure 5.**11**, the results show from the graph also indicates that the path that the tracked human is taking is also a path that is not commonly used since the deviancy counter also has exceeded 10 frames for this video. However, although the result shown is similar to

Figure 5.**9**, this result is a bit more distorted. For example, the results between frame 60 to frame 67, frame 289 to frame 301 and frame 374 to frame 380 is a bit irregular as their output indicates that the tracked human is taking a commonly taken path in the first frame and suddenly change their results to indicate that the same tracked human is taking a not commonly taken path in the next frame.

Such irregularity occurs because the shadow of the tracked human appears during the taking of the video for testing. It was late afternoon when the video was taken and the sun was setting which changed the illumination of the scene and caused shadow to be cast on the wall in the scene which caused certain inaccurate tracking during the testing. The figure below shows the deviant path is taken by the detected human.

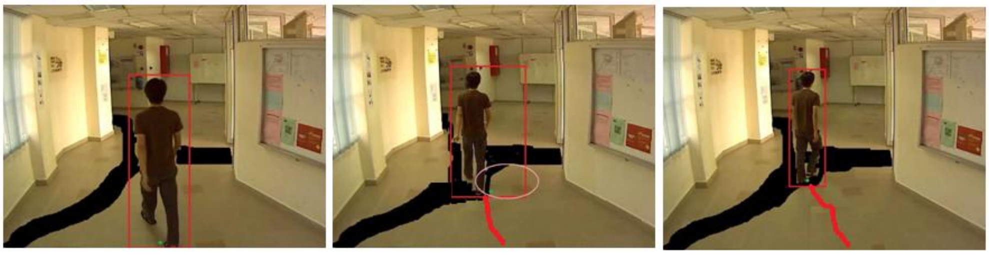


Figure 5.12: Path Taken 2 Scene 1

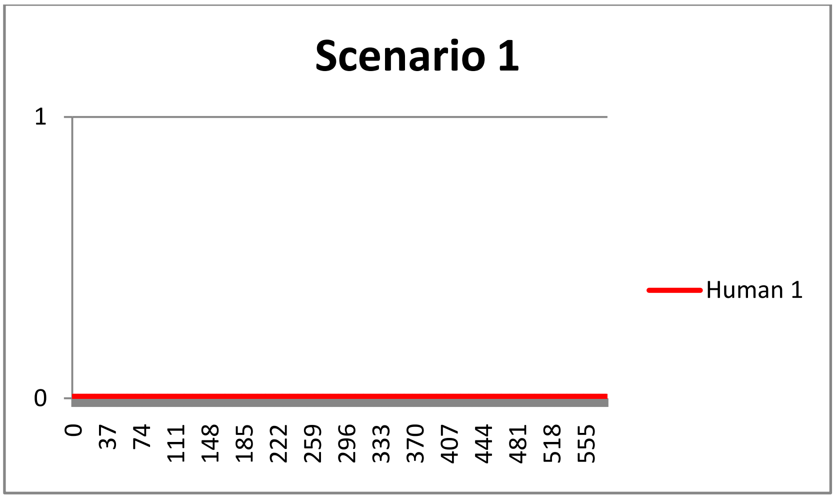


Figure 5.13: Graphical Result 3 Representative for Scene 1

Figure 5.13 shows a graph that indicates the path that is taken by the tracked human in the scene is a path that is commonly taken. The deviancy counter for this video is 0 frames. Thus, based on the deviancy counter and the result shows in the graph, nothing irregular occurs and the result from the graph clearly indicate that this tracked human is taking the path that is commonly taken. This video is also an example of data sets that shows the human that appears in the scene is being tracked accurately and the comparison of the path that is taken by the same tracked human is accurate too. Throughout the first frame till the last frame, the overall performance for this test data is considered very good and accurate. The figure shows the path that is taken by a human taking a normal path.

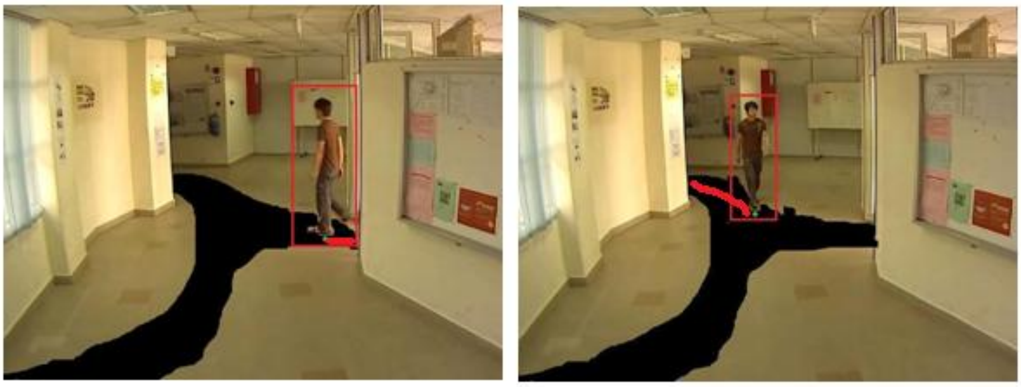


Figure 5.14: Path Taken 3 Scene 1

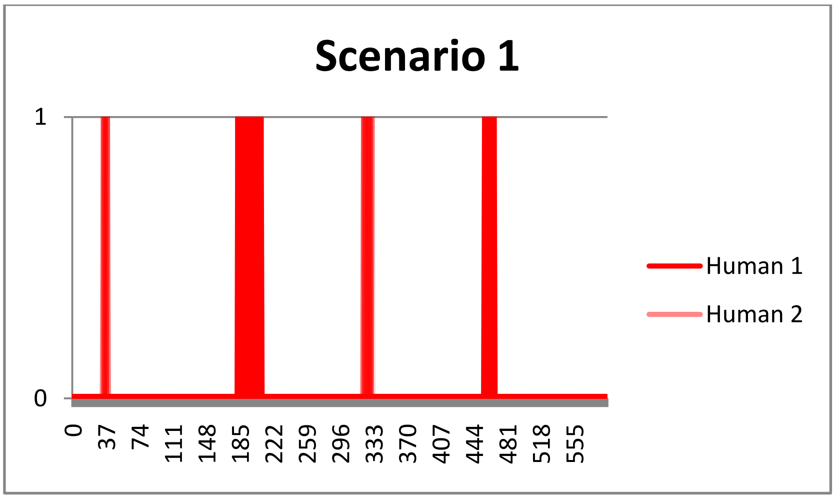


Figure 5.15: Graphical Result 4 Representative for Scene 1

For

Figure 5.**15**, the result shows that the path that is taken by 2 different tracked human that appear in the scene is also a path that is commonly taken. However, the result shows in this graph is a bit different from

Figure 5.**13**. The result in this graph is a bit distorted even though the outcome of the results clearly indicates that the path that is taken by the 2 tracked humans is commonly used. The reason for such distorted results is because of 2 factors.

The first factor is the same as

Figure 5.**11** which is the illumination factor. With 2 people in the scene, the shadow that is cast by the 2 tracked human will greatly affect the accuracy of the human tracking function as the human tracking function has no shadow removal function. Not removing the shadow is one of the limitations of the proposed system.

The second factor would be occlusion. As mentioned in the first factor, due to the sun's position in the late afternoon, shadow are being cast by the 2 tracked human and this shadows will also create contours in the proposed system. Coincidentally, the 2 shadows that are cast overlap each other's shadows and in the proposed system this can be categorized as occlusion as 2 different contours have combined to become one big contour. The tracking function and the human detection function will be greatly affected and thus the irregular results shown in the graph above.

Though the deviancy counter for this video rose above 0 frames due to the external factors as mentioned above, it did not hit 10 frames consecutively and thus this video also shows the tracked humans taking a normal path. The figure below shows the normal path that is taken by the 2 detected humans.

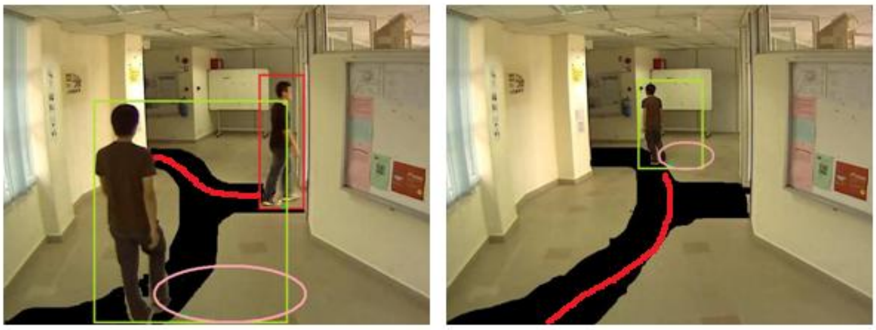


Figure 5.16: Path Taken 4 Scene 1

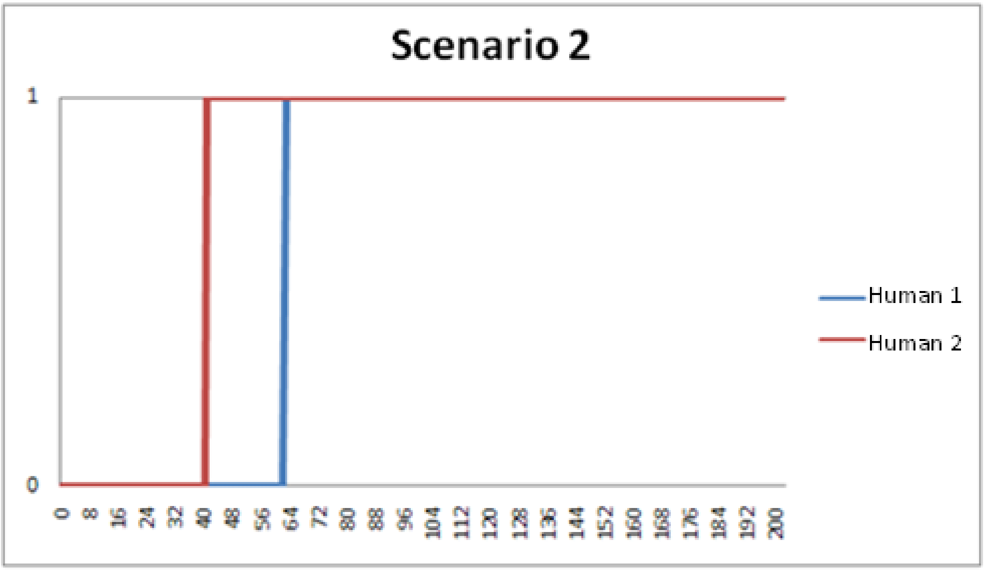


Figure 5.17: Graphical Result 1 Representative for Scene 2

Figure 5.17 show clearly that the two people that are currently being tracked are taking a Deviant Path. The deviancy counter for this video for both tracked human has clearly exceeded 10 frames consecutively. In the beginning of the video, the 2 people that are being tracked are said to be taking a normal path. However, it is without a doubt that the 2 people are actually taking a Deviant Path. The figure below shows the deviant path that is taken by the detected human.

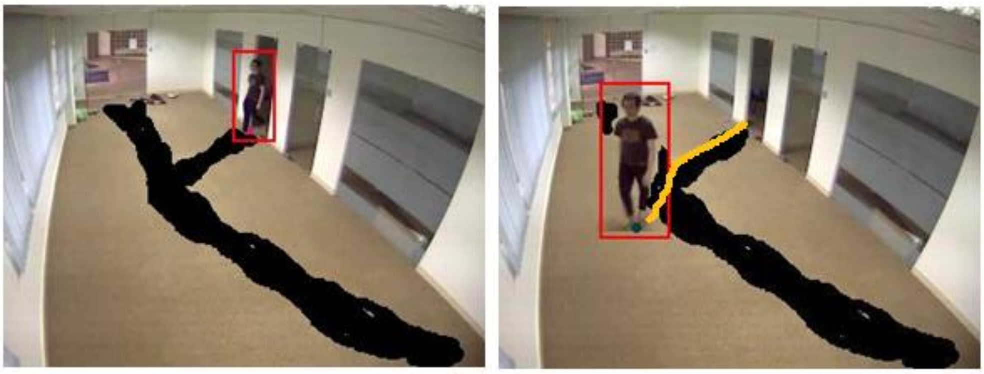


Figure 5.18: Path Taken 1 Scene 2

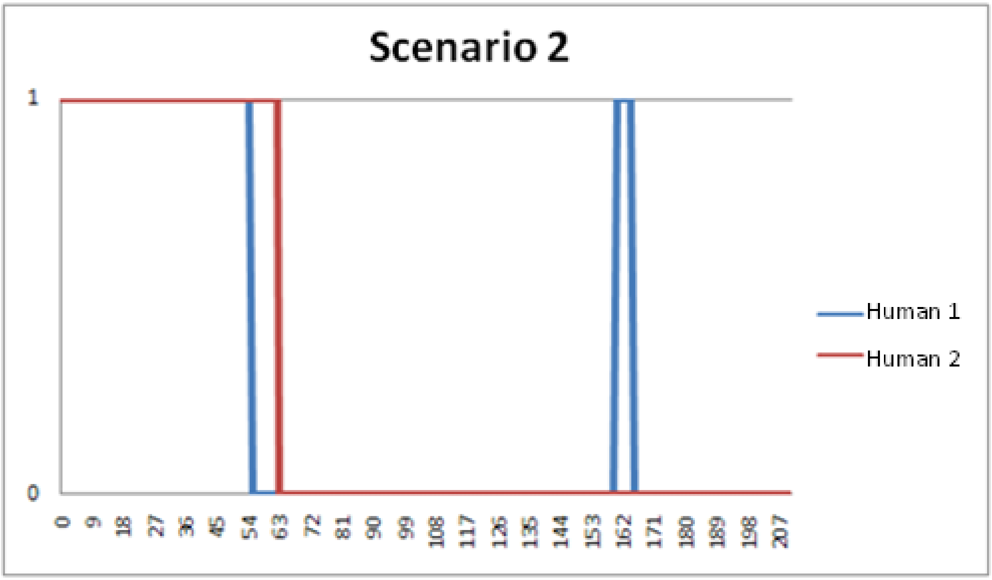


Figure 5.19: Graphical Result 2 Representative for Scene 2

Again, as shown in Figure 5.19, the graph clearly shows that the 2 people that are tracked are taking a Deviant Path due to the deviancy counter exceeding 10 frames consecutively. Though it seems like most of the time the 2 tracked people are taking a normal path, however, this is without a doubt 2 people taking the Deviant Path. At frame number 155 till frame 165, there is a distortion in the results due to the existence of the shadow in the scene. One of the people that are being tracked is walking under a bright light which creates a large enough shadow to cause inaccuracy during the human detection and tracking process. This indirectly also causes the inaccuracy towards the process of getting the accurate coordinates of the tracked human. Thus, the result is slightly distorted. The figure below also shows the human taking a deviant path.

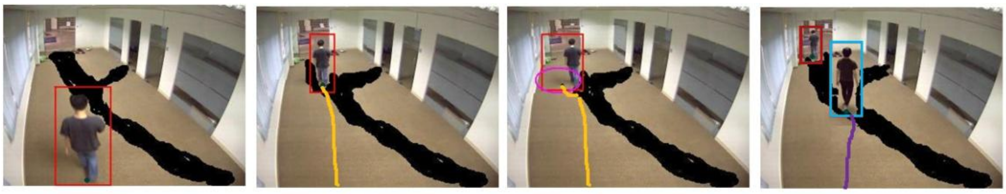


Figure 5.20: Path Taken 2 Scene 2

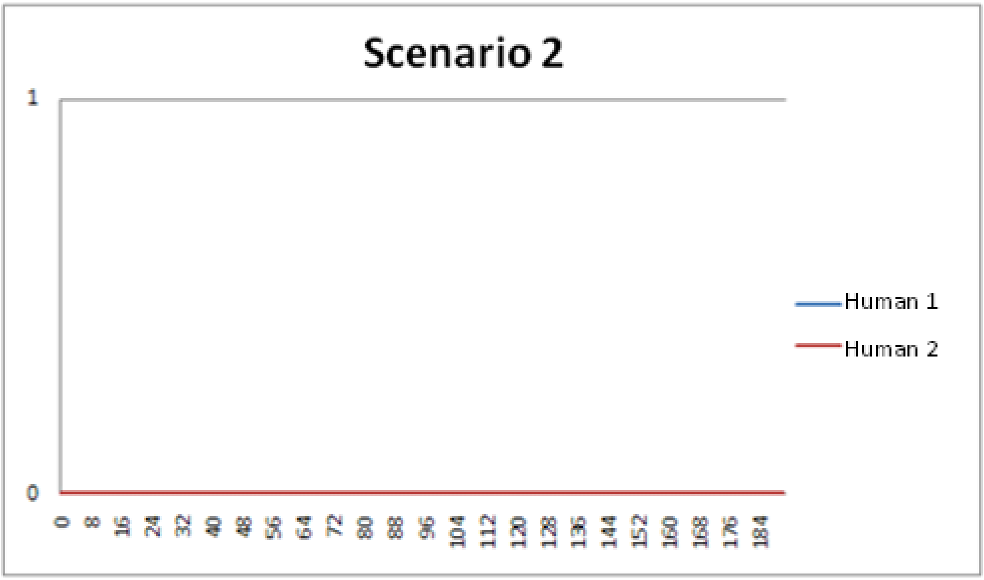


Figure 5.21: Graphical Result 3 Representative for Scene 2

Figure 5.21 shows a graph that indicates the path that are taken by 2 tracked human in the scene is a path that is commonly taken. Even the deviancy counter for this video is at 0 frames. In this graph, nothing irregular occurs and the result from the graph clearly indicate that these 2 tracked human are taking the path that is commonly taken. This video is also an example of data sets that shows the human that appears in the scene are being tracked accurately and the comparison of the path that is taken by the same tracked human is accurate too. Throughout the first frame till the last frame, the overall performance for this test data is considered very good and accurate. The figure below shows the human taking a normal path.

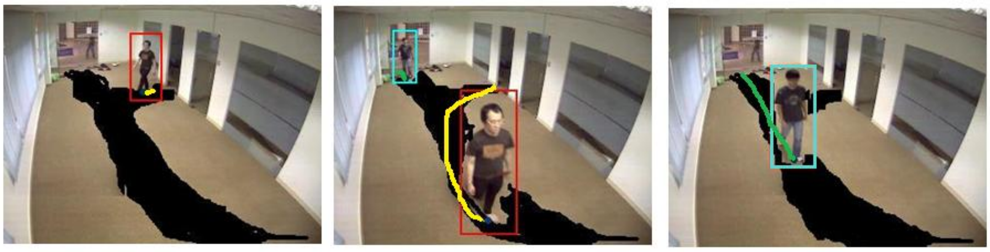


Figure 5.22: Path Taken 3 Scene 2

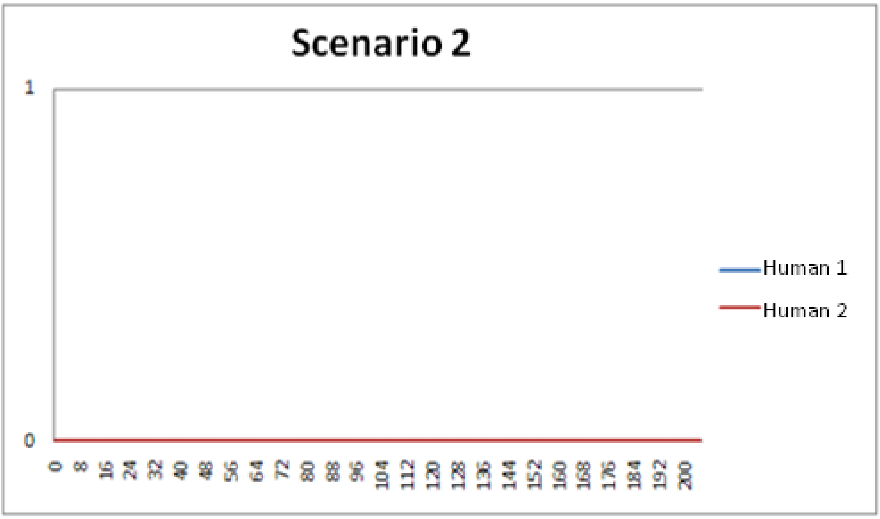


Figure 5.23: Graphical Result 4 Representative for Scene 2

Figure 5.23 shows a graph that indicates the path that are taken by 2 tracked human in the scene is a path that is commonly taken. Similarly as the results previous result, the deviancy counter for this video is also at 0 frames. Thus in this graph, nothing irregular occurs and the result from the graph clearly indicate that these 2 tracked human are taking the path that is commonly taken. This video is also an example of data sets that shows the human that appears in the scene are being tracked accurately and the comparison of the path that is taken by the same tracked human is accurate too. Throughout the first frame till the last frame, the overall performance for this test data is considered very good and accurate.

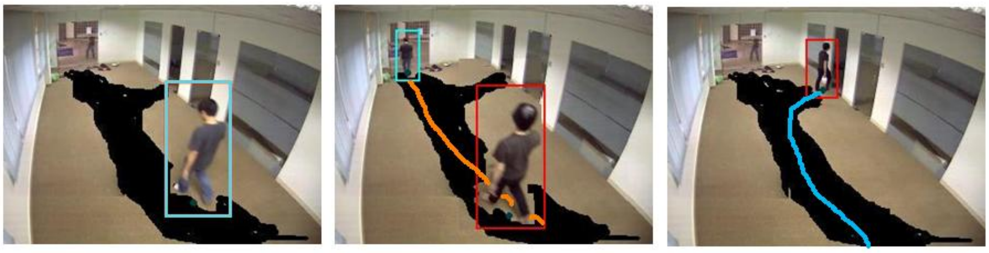


Figure 5.24: Path Taken 4 Scene 2

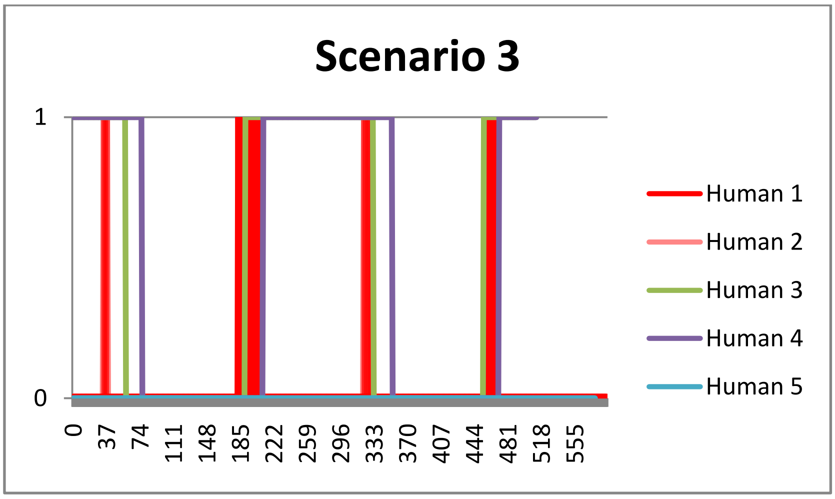


Figure 5.25: Graphical Result 1 Representative for Scene 3

For this video, the lighting of the scene was set to be bright and the result of this video is shown in the graph. With bright lighting, shadows casted by tracked human are detected once in a while causing distortion to the result.

Figure 5.**25** show clearly that the two people that are currently being tracked are taking a Deviant Path while the remaining 3 humans are taking the normal path. The deviancy counter for this video for both human 3 and human 4 has clearly exceeded 10 frames consecutively. Both human 3 and human 4 are deemed to take the Deviant path. The remaining 3 humans however are without a doubt taking the normal path. Although the graphical results for the remaining 3 humans are distorted, it is without a doubt that they are actually taking a Normal Path because their deviancy counter did not exceed 10 frames consecutively. Figure below shows the path taken by the 5 humans in the scene where 2 of them are taking a deviant path while the remaining 3 are taking a normal path.

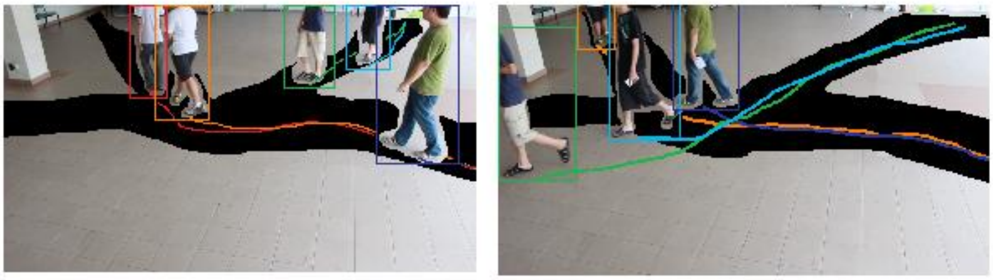


Figure 5.26: Path Taken 1 Scene 3

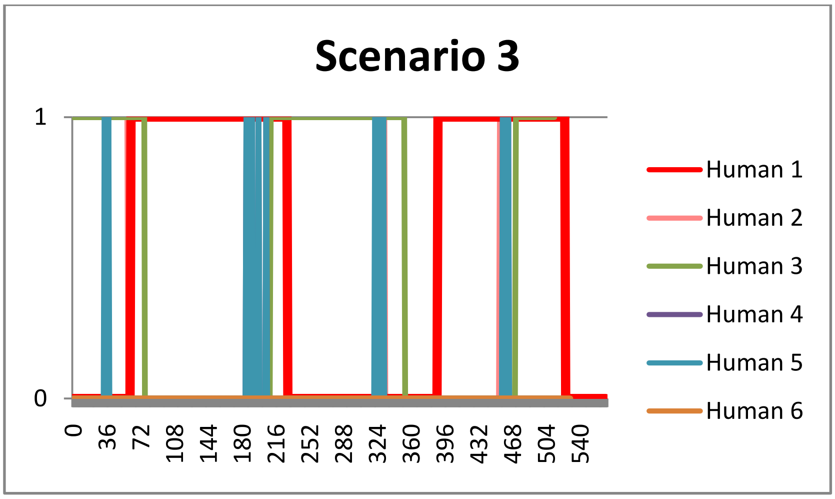


Figure 5.27: Graphical Result 2 Representative for Scene 3

Figure 5.27 shows that Human 1, 2 and 3 are taking a deviant path but human 4, 5 and 6 is taking the normal path. For human 2 and 3, their deviancy counter has exceeded 10 frames consecutively right from the beginning of the video while human 1 is deemed to take the Deviant path only at the latter part of the video. Human 4, 5 and 6 are deemed to take the normal path. There is no distortion of results from human 6 so it is obvious that human 6 is taking the normal path. As for human 4 and 5, they too are deemed taking the normal path because their deviancy counter value did not exceed the value 10 frames consecutively. However, in this video, there is a slight irregularity with the tracker for human 4 and 5. As shown in the Figure 5.28, on the left it is shown that tracker for human 4 and tracker for human 5 in 2 different colours.

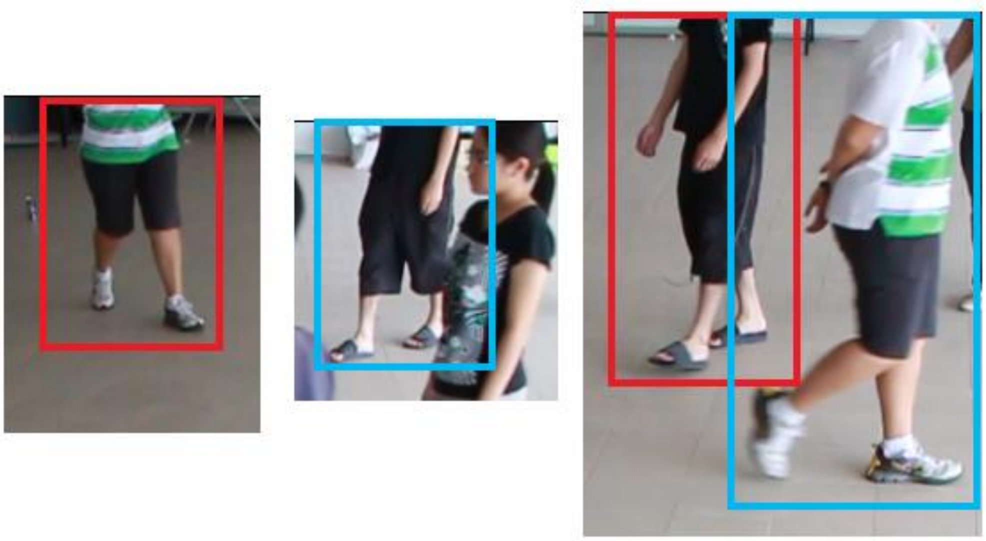


Figure 5.28: Tracker Mismatched

However, once both human 4 and 5 crossed each other and overlapped, their tracker will be switched with each other. The trackers are still functioning and able to continuously track the human. However, only their tracker owners are being switched once they overlapped (cross) each other. Figure below show the path that is taken by the detected human.

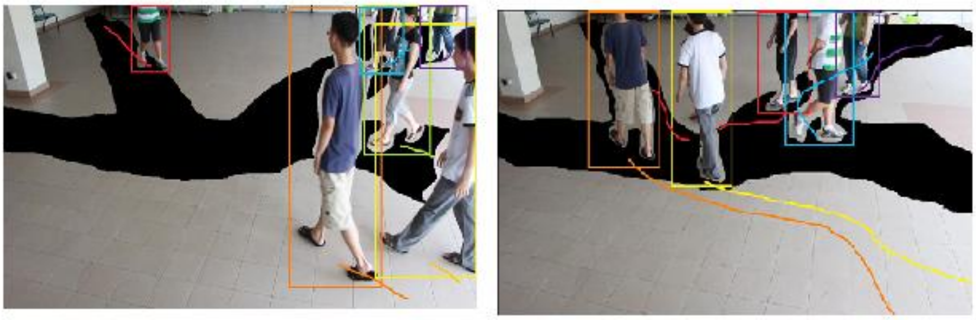


Figure 5.29: Path Taken 2 Scene 3

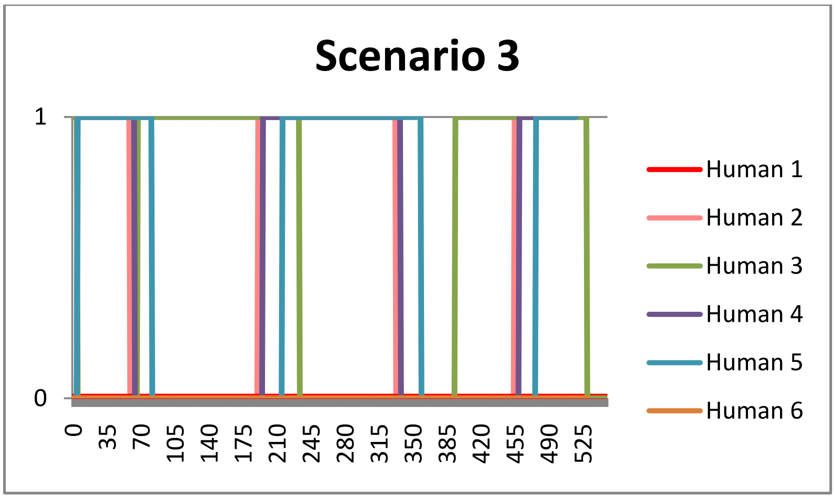


Figure 5.30: Graphical Result 3 Representative for Scene 3

For this video, the scene's lighting was set to a slight dim compare to the other 3 video taken in this scene. Thus, the shadow that is casted by the tracked human is not so noticeable and thus no distortion occurs in this result. Figure 5.30 show that human 2 to 5 is taking the Deviant path while human 1 and 6 is taking the normal path. The deviancy counter for human 2 to 5 has clearly exceeded 10 frames consecutively and thus they are deemed taking the Deviant path while human 1 and 6 with no distortion results shown on the graph are clearly deemed taking the normal path. In this video there are no irregularities at all that occurs and thus the overall performance for this test data is considered very good and accurate.

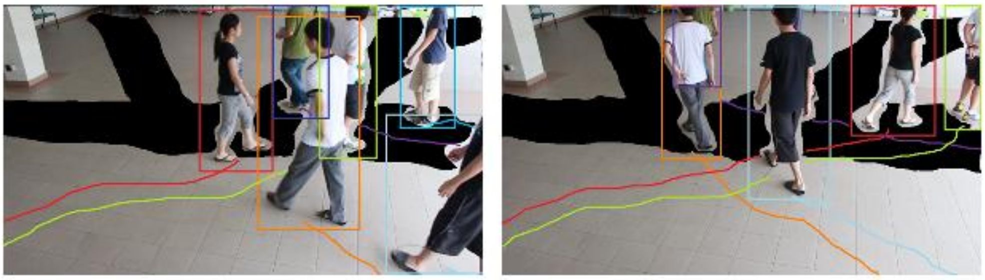


Figure 5.31: Path Taken 3 Scene 3

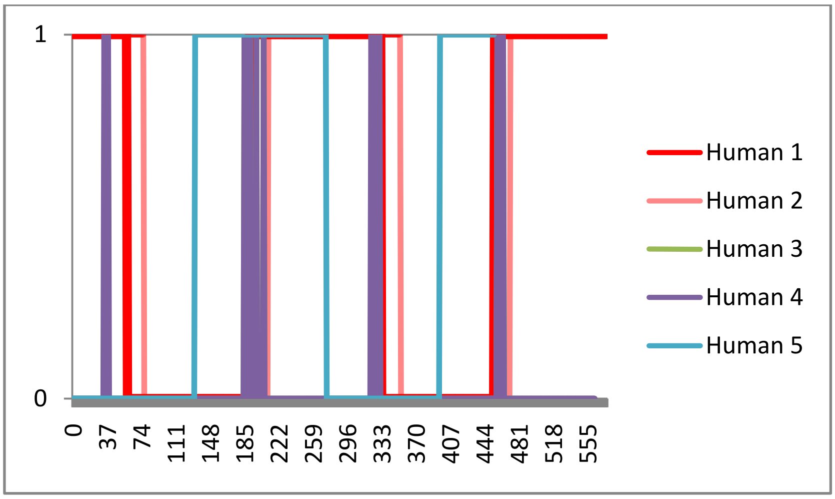


Figure 5.32: Graphical Result 4 Representative for Scene 3

Figure 5.32 shows a graphical result based on the tracking of 5 humans in the scene. Out of the 5 humans, 3 humans were deemed taking the Deviant path while the remaining 2 were deemed taking the normal path. Human 1, 2, and 5 have exceeded their deviancy counter of 10 frames consecutively and thus they are deemed to be taking the Deviant path. Similarly, human 3 and 4 are deemed to be taking the normal path because their deviancy counter did not exceed 10 frames consecutively even though if the results shown on the graph is a bit distorted. Figure below shows the path that is taken by the detected human.

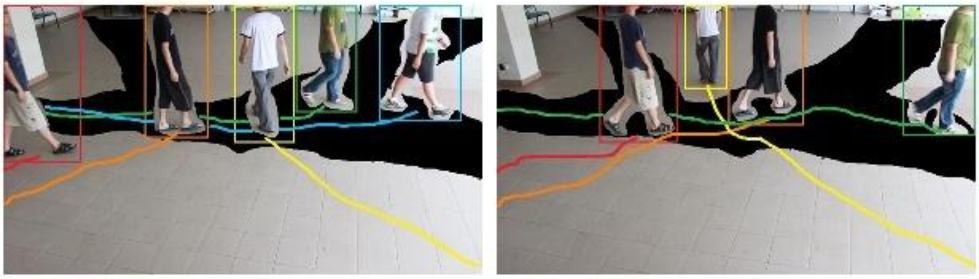


Figure 5.33: Path Taken 4 Scene 3

5.4. Summary of the System Evaluation

To summarize the whole evaluation, the evaluation based on the number of computations clearly shows that the improved Levenshtein Distance method that is used in the proposed system is more efficient than the Levenshtein Distance that is also implementing the Dynamic Time Warping Technique. The number of computations that is reduced is 40.87% and 55.22% of the original total number of computations that is needed to cluster the path. The evaluation of the accuracy of path clustering also shows that the new method is quite accurate with their clustering.

As for the evaluation of the trajectory based activity analysis, the results indicate clearly that the proposed system is capable of indicating which tracked human in the scene is taking a commonly used path and which is not based on the value on the deviancy counter.

5.5. Conclusion

The testing and the discussion of the proposed system will be concluded here and as shown above, the proposed system is made up of many small components. Although they are small and seem insignificant, they still play an important role in the proposed system. If any of the small components fail to function properly, this would affect the proposed system and the overall performance of the proposed system could be reduced greatly.

For the proposed system, unlike most system where the end process is the most essential and important process, the proposed system's most essential and important process lies in the beginning of the process, Module 1 (pre-processing and feature extracting process). If in Module 1, during the preprocessing and feature extracting process, the blob is not obtained correctly, then this would affect the human detection function. If the human detection function cannot detect the human correctly, then this would greatly affect Module 2 (Path Clustering Module) and would also cause the creation of an inaccurate Generic topographical scene description. With an inaccurate Generic topographical scene description, Module 3 (Suspicious Path Detection) will not be able to accurately detect and analyze the path that is taken by the tracked human in the scene. Therefore, if the beginning process cannot function well, the overall performance of the proposed system will be affected greatly like a domino effect. In the next chapter, the summary of the whole research work will be provided together with the definition of the future enhancement about the research work.

CHAPTER 6

FUTURE WORKS AND CONCLUSION

6.1. Introduction

In this chapter, the future enhancement and the conclusion of the research work will be discussed. Due to the time constraint, the enhancement of the proposed system such as increasing the efficiency of the system, improving the robustness of the system, increase the computation speed and improve the system accuracy by removing certain limitations that is imposed on the proposed system is referred to as future enhancement. At the conclusion, a summary of the whole research work of the proposed system will be provided.

6.2. Future Work

The proposed system, “Security Alert through Path Analysis”, is bound by a few limitations. Thus, with these limitations the proposed system is not suitable yet to be deployed and implemented for real life applications. These limitations are necessary for the time being due to the limited time that is available. Therefore, in order to make the proposed system to be suitable for deployment and implementation in the real life applications, the proposed system needs to be enhanced and improved by removing the limitations of the system.

Firstly, the proposed system's human tracking can only tracked human based on the Optical Flow method. This method will used the contour that is obtained by static background subtraction and plot the track point in the contour in order to calculate the optical flow to continuously track the individual that appear in the scene. However, the existence of shadow might cause the contour to be larger than the expected result and thus causing the accuracy of the proposed system to drop. Therefore, with the implementation of the shadow removal, the proposed system's accuracy and the overall system performance will definitely improve.

Secondly, during the tracking of the individuals that is captured on the scene, individuals will most likely occlude one another. This is due to the fact that in real life situations, humans tend to walk in group and the area that requires security surveillance system is usually crowded. Since the proposed system uses Optical Flow to track these individuals, should these individuals occlude one another during the tracking process, errors and inaccurate tracking will happen. An example is shown in Figure 5.28 where 2 humans had their tracker switched with each other. Thus, a function where occlusion of the tracked individuals will still be able to be tracked accurately will be needed in order for the system to be more efficient, accurate and robust.

Thirdly, the proposed system only utilized the deviation from the generic topographical scene description as the only factor to determine whether or not the security should be alerted. With only 1 factor to be used as reference is obviously not applicable to real life situation. In order to improved the proposed system in terms of accuracy and overall performance, the tracked individuals moving speed (velocity) could be included and be used as reference too. The velocity of the individual will be vital in determining whether or not this individual is a potential snatch thief.

Another factor could be included into the proposed system in order to further improve the overall performance. Individuals with ulterior intention usually will walk with a walking trajectory pattern that is different from the norm. By including the trajectory of the moving tracked individual into consideration, the proposed system could then make a better and more accurate judgment as to whether to alert security or not. Ulterior motives such as potential snatch thieves and pickpockets can be better prevented.

With so many factors that can be used as reference in the detection of suspicious behavior, the result through these factors can be affected greatly by the positioning of the camera. The distance, angle and size of the object will be different with cameras at different angle and position. Thus, during the preprocessing of the captured video, a normalization of the angle and camera's position function must be included. In that way, the position and the angle of the camera will not matter and the overall performance and the system's robustness will improve.

The illumination function is the sixth future enhancement. The proposed system can only be used during the daytime and indoor only. The reason is because the proposed system is not capable of solving the illumination problem. If the background that is captured is during the day, then the lighting for the background during the afternoon and night will be different. During the afternoon, the background will be so bright while during the night the background will be very dark. With 3 different backgrounds during the morning, afternoon and night, the results will be different. Thus, with the illumination function included, the system can be more robust.

Finally, the last future enhancement would be a more robust and dynamic path classification. The current path classification method is too sensitive to external factor and the result will be affected by factors such as shadow or the sudden movement of small incident such as the flickering of lights. With a more robust and dynamic path classification, the system would not be so easily affected by external factors.

6.3. Conclusion

As stated in Chapter 1-Introduction, computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information in the forms of decisions while image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. With the advancing technology and the development of many smart system, human’s work load and burden can be reduced and thus saving more valuable human resources that can be redirected to be used in other priorities.

The proposed system is designed for the purpose of detecting and tracking the human that is captured in the scene for monitoring purposes. By monitoring the tracked human, the proposed system can detect any anomaly that is shown in the tracked human’s walking pattern or the path that is taken by them. Through the training process, a normal generic topographical scene description that indicates path that is frequently taken by human will be created and this generic topographical scene description will be used as a reference to decide whether the path that is taken by the tracked human is normal or not. Furthermore, this monitoring function can be develop one step further by adding more decision factors (e.g. velocity and trajectory) so that the proposed system can be implemented in security and monitoring system to improve the quality of the security of a specific area by improving the reaction time during an emergency (e.g. medical emergency, snatch thief prevention, pickpocket prevention).

In order to develop the proposed system, research about similar works done by other researcher has been studied. Works and papers that are written by other researchers are reviewed and summarized in Chapter 2-Literature Review. Through the keen study of other researchers’ work, a general and feasible idea of the method of implementing the proposed system can be obtained and a detailed explanation of the function and techniques used in each module is discussed in Chapter 3-System Implementation. A detail explanation of the contribution and how the method work will be given in Chapter 4-Implemented System Enhancement.

The detailed explanation about the system testing is discussed in Chapter 5-Result and Discussion. Evaluation of the proposed system will be done based on some predefined evaluation mechanisms that are suitable for the proposed system and the results from the evaluation of the proposed system will be studied and analyzed thoroughly. Thus, a summary of the studied and analyzed evaluation will be shown and discussed in Chapter 5-Result and Discussion. A number of 21 videos are being used for the testing process**.**

As shown in Chapter 5-Result and Discussion, through the training process, the normal generic topographical scene description generated as the outcome of the training process will be used as a reference in the testing process of the proposed system. The testing results shows that the proposed system is capable of monitoring humans that appear in the scene for detection humans that are either taking the normal path or deviating from the normal path as stated in the generic topographical scene description that is used as a reference.

The proposed system has achieved all the objectives and was able to finish on time as planned as stated in the Gantt chart in Chapter1-Introduction.

Though the system works well, it is still not yet ready for implementing in the real life situation as there are still limitations in the proposed system as stated in this chapter at section 6.2 Future Work.

Bibliography

Almog, A., Levi, A. and Bruckstein , A. M., 2005. Spatial de-interlacing using dynamic time warping. *International Conference on Image Processing*, 2, pp. 11-13.

Arnold. Wiliem, Madasu, V., Boles, W. and Yarladgadda, P., 2008. Detecting uncommon trajectories. *Proceedings of the Digital Image Computing: Techniques and Applications*, 16, pp. 398-404.

Arnold Wiliem, Madasu, V., Boles, W. and Yarlagadda, P., 2009. A Context Based Approach for Detecting Suspicious Behaviors. *Digital Image Computing: Techniques and Applications*, 5, pp. 146-153.

Bartolini, I., Ciaccia, P. and Patella, M., 2005. WARP: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *Pattern Analysis and Machine Intelligence*, 27 (1), pp. 142-147.

Bashir, F.I., Khokhar, A.A. and Schonfeld, D., 2003. Segmented trajectory based indexing and retrieval of video data. *International Conference on Image Processing*, 3 (2), pp. 623-626.

Bhaskar Chakraborty, Ognjen Rudovic and Jordi Gonzalez, 2008. View-Invariant Human Body Detection with Extension to Human Action Recognition using Component-Wise HMM of Body Parts. *International Conference on Automatic Face & Gesture Recognition*, 8, pp. 1-6.

Bhuvaneswari, K. and Rauf A.H., 2009. Edgelet Based Human Detection and Tracking by Combined Segmentation and Soft Decision. *International Conference on Control, Automation, Communication and Energy Conservation*, 1, pp. 1-6.

Boiman, O. and Irani, M., 2005. Detecting irregularities in images and in video. *International Journal on Computer Vision*, 74 (1), pp. 17-31.

Boudraa, A.O., Cexus, J.C., Groussat, M. and Brunagel, P., 2008. An energy-based similarity measure for time series. *European Association for Signal Processing*, 26, pp. 280-290.

Brown, J. C., Hodgins-Davis, A. and Miller, P. J. O., 2006. Classification of vocalizations of killer whales using dynamic time warping. *The Journal of the Acoustical Society of America*, 122 (2), pp. 1201-1207.

Chan, A.B., Mahadevan, V. and Vasconcelos, N., 2010. *Generalized Stauffer-Grimson Background Subtraction for Dynamic Scenes*. Springer-Verlag:Machine Vision and Applications.

Chang, W. and Shin, J., 2007. Modified Dynamic Time Warping for stroke-based on-line signature verification. *International Conference on Document Analysis and Recognition*, 2, pp. 724-728.

Che Fung and Nicholas Jerrat, 2000. A Neural Network Based Intelligent Intruders Detection and Tracking System using CCTV Images. *International of Technical Conference*, 2, pp. 409-414.

Cheng, L. and Gong, M., 2009. Realtime Background Subtraction from Dynamic Scenes. *International Conference on Computer Vision*, 12, pp. 2066-2073.

Choo, C.Y., Lee, K., See, H.Q., Tan, Z.J. and Khor, S.W., 2010. Pedestrian Detection with Image Segmentation and Virtual Mask. *International Conference on Computer Science and Information Technology*, 4, pp. 23-27.

Comaniciu, D., Ramesh, V. and Meer, P., 2003. Kernel-Based Object Tracking. *Pattern Analysis and Machine Intelligence*, 25 (5), pp. 564-577.

Csaba Beleznai Horst Bischof, 2009. Fast Human Detection in Crowded Scenes by Contour Integration and Local Shape Estimation. *International Conference on Computer Vision and Pattern Recognition*, 10, pp. 2246-2253.

Dalal, N. and Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. *International Conference on Computer Vision and Pattern Recognition*, 1, pp. 886-893.

Daniel Barbara and Maurizio Filippone, 2008. Detecting Suspicious Behavior in Surveillance Images. *International Conference on Data Mining Workshops*, 8, pp. 891-900.

Denman, S., Fookes, C. and Sridharan, S., 2009. Improved Simultaneous Computation of Motion Direction and Optical Flow for Object Tracking. *Digital Image Computing: Techniques and Applications*, 5, pp. 175-182.

Dimitrios Makris and Tim Ellis, 2002. Path Detection in Video Surveillance. *Journal of Image and Vision Computing*, 20 (12), pp. 895-903.

Ding, W., Li, H., Jiang, Z. and Li, X., 2009. Unsupervised Spatio-Temporal Multi-Human Detection and Recognition in Complex Scene. *International Congress on Image and Signal Processing*, 1, pp. 1-5.

Duda, R. O., Hart, P. E. and Stork, D. G., 2000, *Pattern Classification*, 1st ed. A Willey-Interscience Publication.

Enzweiler, M. and Gavrila, D.M., 2009. Monocular Pedestrian Detection: Survey and Experiments. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 31 (12), pp. 2179-2195.

Faloutsos, C., 1996. *Searching Multimedia Databases by Content*. 1st ed. Kluwer Academic Publishers.

Faundez-Zanuy, M., 2007. On-line signature recognition based on VQ-DTW. *Pattern Recognition*, 40 (3), pp. 981-992.

Freer, J A, Beggs, B J, Femandez-Canque, H L, Chevrier, F. and Goryashko, A., 1997. Automatic intruder Detection incorporating Intelligent Scene Monitoring With Video Surveillance. *European Conference on Security and Detection*, 10, pp. 109-113.

Fu, Z., Hu, W. and Tan, T., 2005. Similarity Based Vehicle Trajectory Clustering and Anomaly Detection. *International Conference on Image Processing*, 2, pp. 602-605.

Gupta, A. and Davis, L. S., 2007. Objects in action: An approach for combining action understanding and object perception. *International Conference on Computer Vision and Pattern Recognition*, 10, pp. 1-8.

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

Hua, Z., Jianbo, S. and Visontai, M., 2004. Detecting unusual activity in video. *International Conference on Computer Vision and Pattern Recognition*, 2, pp. 819-826.

Hui-Xing Jia and Yu-Jin Zhang, 2007. Fast Human Detection by Boosting Histograms of Oriented Gradients. *International Conference on Image and Graphics*, 13, pp. 683-688.

Imran N. Junejo and Hassan Foroosh, 2007. Trajectory Rectification and Path Modeling for Video Surveillance. *International Conference on Computer Vision*, 11, pp. 1-7.

Ince, S. and Konrad, J., 2008. Occlusion-Aware Optical Flow Estimation. *IEEE Trans. On Image Processing*, 17 (8), pp. 1443-1451.

Jang, J. S. R. and Lee, H.-R., 2008. A general framework of progressive filtering and its application to query by singing/humming. *IEEE Transactions on Audio, Speech, and Language Processing*, 16 (2), pp. 350-358.

Javed, O. and Shah, M., 2002. Tracking and Object Classification for Automated Surveillance. *European Conference on Computer Vision*, 2353, pp. 343-357.

Jorge Baranda, Vincent Jeanne and Ralph Braspenning, 2008. Efficiency Improvement of Human Body Detections With Histograms of Oriented Gradients. *International Conference on Distributed Smart Cameras*, 2, pp. 1-9.

Junejo, I., Javed, O. and Shah, M., 2004. Multi Feature Path Modeling for Video Surveillance. *International Conference on Pattern Recognition*, 2, pp. 716-719.

Kale, A., Cuntoor, N., Yegnanarayana, B., Rajagopalan, A. N. and Chellappa, R., 2004. Optical and Digital Techniques for Information Security. *Springer-Verlag Gait-Based Human Identification Using Appearance Matching*, 97 (6), pp. 1128-1148.

Kanhere, N.K., Pundlik, S.J. and Birchfield, S.T., 2005. Vehicle Segmentation and Tracking from a Low-Angle Off-Axis Camera. *International Conference on Computer Vision and Pattern Recognition*, 2, pp. 1152-1157.

Katz, I. and Aghajan, H., 2007. A case study: Exploring relationship between context and pose, 2nd ed. Cognitive Systems and Interactive Sensors.

Keogh, E.J. and Pazzani, M.J., 2000. Scaling up Dynamic Time Warping for Datamining Application. *International Conference on Knowledge discovery and data mining*, 6, pp. 285-289.

Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L., 2005. Real-time Foreground-Background Segmentation Using Codebook Model. *Real-Time Imaging: Special Issue on Video Object Processing*, 11, pp. 172-185.

Kim, Z. 2008. Real Time Object tracking based on Dynamic Feature Grouping with Background Subtraction. *International Conference on Computer Vision and Pattern Recognition*, 8, pp. 1-8.

Lee, K., Choo, C.Y., See, H.Q., Tan, Z.J. and Lee, Y.L., 2010. Human Detection using Histogram of Oriented Gradients and Human Body Ratio Estimation. *International Conference on Computer Science and Information Technology*, 4, pp. 18-22.

Legrand, B., Chang, C. S., Ong, S. H., Neo, S. Y. and Palanisamy, N., 2007. Chromosome classification using dynamic time warping. *Pattern Recognition Letters*, 29 (3), pp. 215-222.

Lian, X., Zhang, T. and Liu, Z., 2010. A Novel Method on Moving-Objects Detection Based on Background Subtraction and Three Frames Differencing. *International Conference on Measuring Technology and Mechatronics Automation*, 1, pp. 252-256.

Lijing Chang and Yingli Liang, 2010. Motion Human Detection Based on Background Subtraction. *International Workshop on Education Technology and Computer Science*, 1, pp. 284-287.

Lopez, L. E. M., Elias, R. P. and Tavira, J. V., 2007. Face localization in color images using dynamic time warping and integral projections. *International Joint Conference on Neural Networks*, 12, pp. 892-896.

Mahadevan, V. and Vasconcelos, N., 2008. Background Subtraction in Highly Dynamic Scenes. *International Conference on Computer Vision and Pattern Recognition*, 7, pp. 1-6.

Makris, D. and Ellis, T., 2005. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics*, 35 (3), pp. 397-408.

Marzal, A., Palazon, V. and Peris, G., 2005. Contour-based shape retrieval using Dynamic Time Warping. *International Conference of the Spanish Association for Artificial Intelligence*, 19, pp. 190-199.

McHugh, J.M., Konrad, J., Saligrama, V. and Jodoin, P.M., 2009. Foreground-Adaptive Background Subtraction. *IEEE Signal Processing Letters*, 16 (5), pp. 390-393.

Morency, L.P., Sidner, C., Lee, C. and Darrell, T., 2006. The role of context in head gesture recognition. *National Conference on Artificial Intelligence*, 171 (8), pp. 568-585.

Morris, B. T. and Trivedi, M. M., 2008. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18 (8), pp. 1114-1127.

Muscillo, R., Conforto, S., Schmid, M., Caselli, P. and D'Alessio, T., 2007. Classification of motor activities through derivative dynamic time warping applied on accelerometer data. *International Conference of Engineering in Medicine and Biology Society*, 10, pp. 4930-4933.

Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, and Yozo Fujino, 2007. Learning Motion Patterns and Anomaly Detection by Human Trajectory analysis. *International Conference on Systems, Man and Cybernetics*, 6, pp. 498-503.

Navneet Dalal and Bill Triggs, 2005. Histograms of Oriented Gradient for Human Detection. *International Conference on Computer Vision and Pattern Recognition*, 1, pp. 886-893.

Neil Johnson and David Hogg, 2000. Learning the Distribution of Object Trajectories for Event Recognition. *British Machine Vision Conference*, 14 (8), pp. 609-615.

Nguyen Duc Thanh, Philip Ogunbona and Wanqing Li, 2009. Human Detection Based on Weighted Template Matching. *International Conference on Multimedia and Expo*, 7, pp. 634-637.

Niels, R. and Vuurpijl, L., 2005. Using Dynamic Time Warping for intuitive handwriting recognition. *International Conference of Graphonomics Society Advances in Graphonomics*, 5, pp. 245-251.

Pal, A., Schaefer, G. and Celebi, M.E., 2010. Robust Codebook-Based Video Background Subtraction. *International Conference on Acoustics Speech and Signal Processing*, 10, pp. 1146-1149.

Piciarelli, C. and Foresti, G. L., 2006. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27 (15), pp. 1835-1842.

Porikli, F.M. and Haga, T., 2004. Event Detection by Eigenvector Decomposition Using Object and Frame Features. *International Conference on Computer Vision and Pattern Recognition Workshop*, 7, pp. 114-121.

Porikli, F., Tuzel, O. and Meer, P., 2006. Covariance Tracking Using Model Update Based on Lie Algebra. *International Conference on Computer Vision and Pattern Recognition*, 1, pp. 728-735.

Sakoe, H. and Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26 (1), pp. 43-49.

Sakurai, Y., Yoshikawa, M. and Faloutsos, C., 2005. FTW: fast similarity search under the time warping distance. *Symposium on Principles of database systems*, 7, pp. 326-337.

Sang-Hyun, O., Jin-Suk, K., Yung-Cheol, B., Gyung-Leen, P. and Sang- Yong, B., 2005. Intrusion detection based on clustering a data stream. *International Conference on Software Engineering Research, Management and Applications*, 10, pp. 220-227.

Shaopeng Tang and Satoshi Goto, 2009. Human Detection Using Motion and Appearance based Feature. *International Conference on Information, Communications and Signal Processing*, 3, pp. 1-4.

Shaopeng Tang and Satoshi Goto, 2009. Partially Occluded Human Detection by Boosting SVM. *International Colloquium on Signal Processing & Its Applications*, 14, pp. 224-227.

Shaopeng Tang and Satoshi Goto, 2010. Histogram of Template for Human Detection. *International Conference on Acoustics Speech and Signal Processing*, 10, pp. 2186-2189.

Simon Fong and Zhuang Yan, 2007. A security Model For Detecting Suspicious Patterns in Physical Environment. *International Symposium on Information Assurance and Security*, 11, pp. 221-226.

Simone Calderara, Rita Cucchiara and Andrea Prati, 2007. Detection of Abnormal Behaviors Using A Mixture of Von Mises Distributions. *IEEE Conference on Advanced Video and Signal Based Surveillance,* 4, pp. 141-146.

Simone Calderara, Rita Cucchiara and Andrea Prati, 2008. Using Circular Statistics for Trajectory Shape Analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 4, pp. 1-8.

Stauffer, C. and Grimson, W.E.L., 1999. Adaptive background mixture models for real-time tracking. *International Conference on Computer Vision and Pattern Recognition*, 2, pp. 156-162.

Stauffer, C. and Grimson, E.., 2000. Learning patterns of activity using real-time tracking. *Pattern Analysis and Machine Intelligence*, 22 (8), pp. 747-757.

Tang, S. and Goto, S., 2009. Human Detection Using Motion and Appearance based Feature. *International Conference on Information, Communication and Signal Processing*, 10, pp. 1-4.

Toni M. Rath and R. Manmatha, 2003. Lower-Bounding of Dynamic Time Warping Distances for Multivariate Time Series. *International Conference on Machine Learning and Computing*, 3, pp. 245-249.

Toyama, K., Krumm, J., Brumitt, B. and Meyers, B., 1999. Wallflower: Principles and Practices of Background Maintenance. *International Conference on Computer Vision*, 1, pp. 255-261.

Ujjwal Das Gupta, Vinary Menon and Uday Babbar, 2010. Detecting the Number of Clusters During Expectation-Maximization Clustering Using Information Criterion. *International Conference on Machine Learning and Computing*, 8, pp. 169-173.

Vaswani, N., Roy Chowdhury, A. and Chellappa, R., 2003. Activity recognition using the dynamics of the configuration of interacting objects. *International Conference on Computer Vision and Pattern Recognition*, 2, pp. 633–640.

Vega-Lopez, I. F. and Moon, B., 2006. Quantizing time series for efficient similarity search under time warping. *International Conference on Advances in computer science and technology*, 2, pp. 334-339.

Veltkamp, R. C., 2001. Shape matching: similarity measures and algorithms. *International Conference on Shape Modeling and Applications*, 7, pp. 188-197.

Vlachos, M., Kollios, G. and Gunopulos, D., 2002. Discovering Similar Multidimensional Trajectories. *International Conference on Data Engineering*, 9, pp. 673-684.

Vladimir N. V., 1995. Support-Vector Networks. *Journal of Machine Learning archive*, 20 (3), pp. 273-297.

Wang, J. and Yagi, Y., 2008. Switching Local and Covariance Matching for Efficient Object Tracking. *International Conference on Pattern Recognition*, 17, pp. 1-4.

Wang, Z., Xu, H., Sun, L. and Yang, S., 2009. Background Subtraction in Dynamic Scenes with Adaptive Spatial Fusing. *IEEE International Workshop on Multimedia Signal Processing*, 5, pp. 1-6.

Weber, R., Schek, H.-J. and Blott, S., 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *International Conference on Very Large Data Bases*, 5, pp. 194-205.

Wei, L., Keogh, E., Herle, H. V. and Mafra-Neto, A., 2005. Atomic wedgie: Efficient query filtering for streaming times series. *International Conference on Data Mining*, 5, pp. 490-497.

Xia, Y., Ning, S. and Shen, H., 2010. Moving Targets Detection Algorithm Based on Background Subtraction and Frames Subtraction. *International Conference on Industrial Mechatronics and Automation*, 1, pp. 122-125.

Xie, D., Hu, W., Tan, T. and Peng, J., 2004. A Multi-Object Tracking System for Surveillance Video Analysis. *International Conference on Pattern Recognition*, 4, pp. 767-770.

Xu, Y.W., Cao, X.B. and Qiao, H., 2007. Pedestrian Detection with Local Feature Assistant. *IEEE International Conference on Control and Automation*, 9, pp. 1542-1547.

Yi Feng Zhou and Jim P.Y. Lee, 2010. A Modified Expectation Maximization Algorithm for Maximum Likelihood Direction of arrival Estimation. *Asilomar Conference on Signals, Systems and Computers*, 1, pp. 613-617.

Yue, Z., Yue, Z., Shuicheng, Y. and Huang, T. S., 2007. Detecting anomaly in videos from trajectory similarity analysis. *International Conference on Multimedia and Expo*, 87, pp. 1087-1090.

Zhang Zhang, Kaiqi Huang and Tieniu Tan, 2006. Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. *International Conference on Pattern Recognition*, 3, pp. 1135-1138.

Zhong, B., Liu, S., Yao, H. and Zhang, B., 2009. Multi-Resolution Background Subtraction for Dynamic Scenes. *International Conference on Image Processing*, 23, pp. 3193-3196.

Zhong, J. and Sclarroff, S., 2003. Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter. *International Conference on Computer Vision*, 1, pp. 44-50.

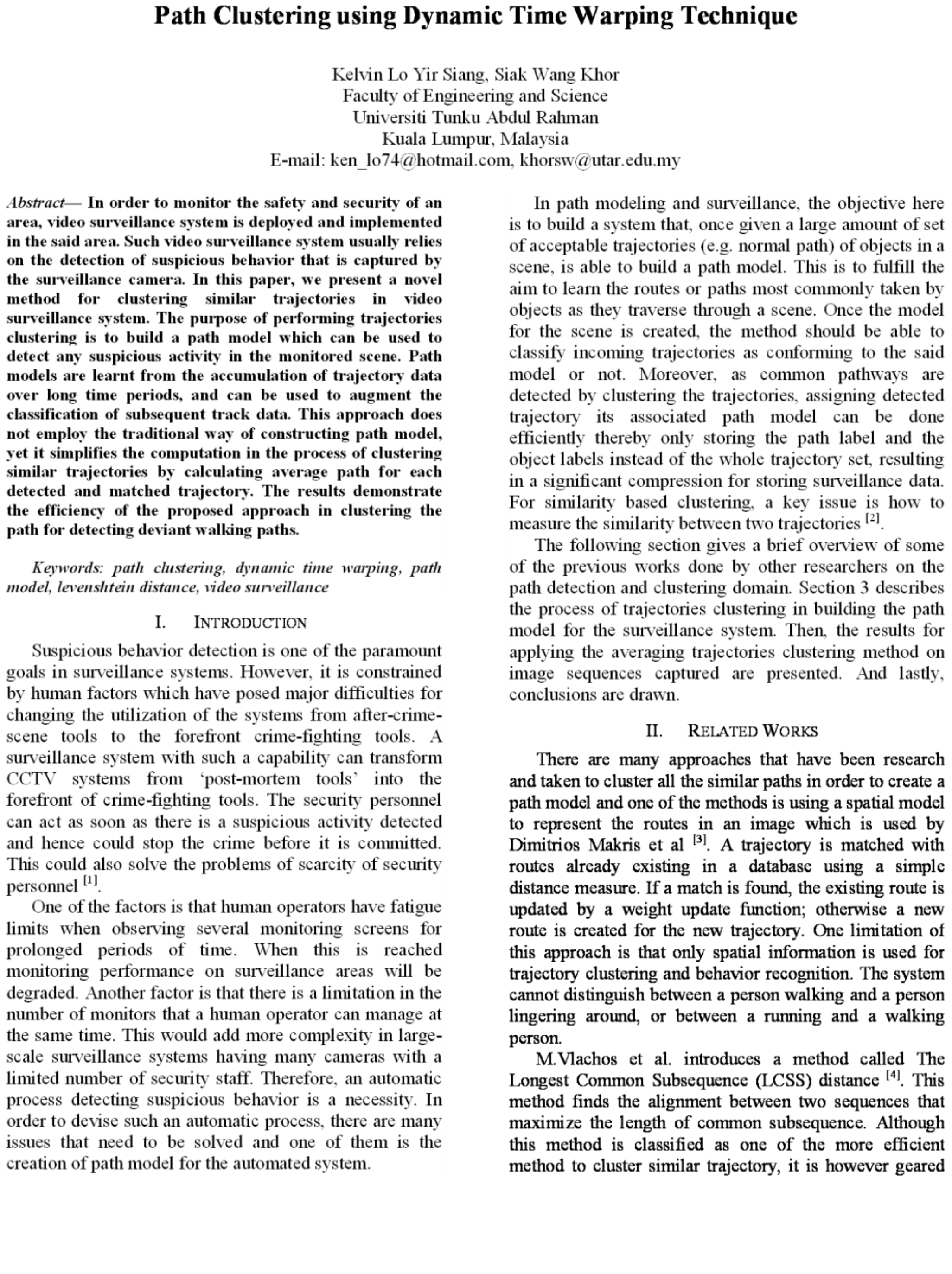
Zhou, M., M. and Wong, H., 2007. Boundary-based lower-bound functions for Dynamic Time Warping and their indexing. *International Conference on Data Engineering*, 23, pp. 1307-1311.

Zhou, Q. and Aggarwal, J.K., 2006. Object Tracking in an Outdoor Environment using Fusion of Features and Cameras. *International Conference on Image and Vision Computing*, 24 (11), pp. 1244-1255.

Zhu, Y. and Shasha, D., 2003. Warping indexes with envelope transforms for query by humming. *Special Interest Group on Management Of Data*, 6, pp. 181-192.

Appendix A

PUBLICATION (International Conference on Networked Computing and Advanced Information Management) – PATH CLUSTERING USING DYNAMIC TIME WARPING TECHNIQUE



Path Clustering using Dynamic Time Warping Technique

Kelvin Lo Yir Siang, Siak Wang Khor
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman
Kuala Lumpur, Malaysia
E-mail: ken_lo74@hotmail.com, khorsw@utar.edu.my

Abstract— **In order to monitor the safety and security of an area, video surveillance system is deployed and implemented in the said area. Such video surveillance system usually relies on the detection of suspicious behavior that is captured by the surveillance camera. In this paper, we present a novel method for clustering similar trajectories in video surveillance system. The purpose of performing trajectories clustering is to build a path model which can be used to detect any suspicious activity in the monitored scene. Path models are learnt from the accumulation of trajectory data over long time periods, and can be used to augment the classification of subsequent track data. This approach does not employ the traditional way of constructing path model, yet it simplifies the computation in the process of clustering similar trajectories by calculating average path for each detected and matched trajectory. The results demonstrate the efficiency of the proposed approach in clustering the path for detecting deviant walking paths.**

Keywords: path clustering, dynamic time warping, path model, levenshtein distance, video surveillance

I. Introduction

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform CCTV systems from 'post-mortem tools' into the forefront of crime-fighting tools. The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel [1].

One of the factors is that human operators have fatigue limits when observing several monitoring screens for prolonged periods of time. When this is reached monitoring performance on surveillance areas will be degraded. Another factor is that there is a limitation in the number of monitors that a human operator can manage at the same time. This would add more complexity in large-scale surveillance systems having many cameras with a limited number of security staff. Therefore, an automatic process detecting suspicious behavior is a necessity. In order to devise such an automatic process, there are many issues that need to be solved and one of them is the creation of path model for the automated system.

In path modeling and surveillance, the objective here is to build a system that, once given a large amount of set of acceptable trajectories (e.g. normal path) of objects in a scene, is able to build a path model. This is to fulfill the aim to learn the routes or paths most commonly taken by objects as they traverse through a scene. Once the model for the scene is created, the method should be able to classify incoming trajectories as conforming to the said model or not. Moreover, as common pathways are detected by clustering the trajectories, assigning detected trajectory its associated path model can be done efficiently thereby only storing the path label and the object labels instead of the whole trajectory set, resulting in a significant compression for storing surveillance data. For similarity based clustering, a key issue is how to measure the similarity between two trajectories [2].

The following section gives a brief overview of some of the previous works done by other researchers on the path detection and clustering domain. Section 3 describes the process of trajectories clustering in building the path model for the surveillance system. Then, the results for applying the averaging trajectories clustering method on image sequences captured are presented. And lastly, conclusions are drawn.

II. Related Works

There are many approaches that have been research and taken to cluster all the similar paths in order to create a path model and one of the methods is using a spatial model to represent the routes in an image which is used by Dimitrios Makris et al [3]. A trajectory is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function; otherwise a new route is created for the new trajectory. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person.

M.Vlachos et al. introduces a method called The Longest Common Subsequence (LCSS) distance [4]. This method finds the alignment between two sequences that maximize the length of common subsequence. Although this method is classified as one of the more efficient method to cluster similar trajectory, it is however geared more towards recognizing shape similarity rather than path trajectory [5].

F.I.Bashir et al. proposed a method called the Principle Components Analysis (PCA) where a trajectory is first represented as a 1-D signal by concatenating the x- and the y-projections [6]. Then the signal is converted into the first few PCA coefficients. The trajectory similarity is computed as the Euclidean distance between the PCA coefficients. There is a disadvantage to this method. It is brittle to the distortion and other noise. However, it can cluster similar trajectory in a short amount of time.

Hausdorff Distance is a method that is used by I.Junejo et al. to cluster similar trajectory [7]. The method get the longest distance among the two shortest distances between two coordinates. The advantage of this method is that it can get the shortest distance after comparing with all possible possibilities. However, this method is unable to properly recognize the direction of the trajectory. The same path but different direction is the same before this method.

F.M. Porikli et al. mentioned that using Hidden Markov Model-Based Distance is a viable solution to cluster similar trajectory [8]. By comparing the likelihood of the trajectories to their own model, similar trajectory will be clustered together. By using probabilities, it is indeed an efficient method. However, due to limited number of training data, it will most likely suffer from over-fitting.

Another approach of clustering similar trajectory would be using the Euclidean Distance Method that is used by Z.Fu et al [9]. By getting all the coordinates of the similar trajectory, a simple calculation will be computed and the median will be plotted out. The advantage of using this method lies with the simplicity of the calculation. However, because the calculation is simple, it is also computational extensive and thus it is also a disadvantage of using this method.

One clustering approach that is implemented by Ujjwal Das Guptaet al. is to assume a Gaussian Mixture Model and implementing the X-Means algorithm [10]. In this model, the input data is assumed to have been generated by selecting any one of the K Gaussian distributions. Each cluster is thus represented by a single distribution. The major problem that is faced by using the X-Means is that is assumes an identical spherical Gaussian of the data and thus it tends to overfit data in elliptical clusters. The advantages of using this algorithm is that can optimally detect the clusters for a chosen value of K.

Neil Johnson and David Hogg present a statistically based model of object trajectories which is learnt from image sequences as a viable solution to cluster similar trajectory and create a path model [11]. Trajectory data is supplied by a tracker using Active Shape Models, from which a model of the distribution of typical trajectories is learnt. From the sequence of the image shown during the training stage, the trajectories of the tracked individual can be obtained. However, because the amount of data is too large, a large storage is needed.

Once a trajectory of a moving object is obtained, it is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function. One limitation of this approach is that only spatial information is used for trajectory clustering.

I. Proposed Methodology

In order to determine the individual's status based on "Deviation from Normal Path" factor, a normal path model that is created through training is needed. The normal path model will be applied to the system as based and any path that is taken by the monitored individual that is deviated from the normal path that is already defined in the normal path model will be deemed as suspicious.

Clustering of the similar trajectory is being done by using Dynamic Time Warping (DTW). The DTW algorithm uses a dynamic programming technique to solve this problem. The reason this method is being used is because it is not computational intensive and it can cluster similar trajectory at a very short amount of time. Furthermore, should 2 path with different numbers of points will still be able to be clustered together to form a normal path. The basic requirement for Dynamic Time Warping is the coordinates that is obtained from both the clustered path and also the newly obtained path. We will calculate the distance of each coordinates from both path and each coordinates will be saved in an array and thus 2 arrays are obtained.

We will be using an improved version of Levenshtein Distance to compare both coordinates. Using this distance has 2 advantages. The first advantage will be both path does not need to have the same amount of points to be compared and the second advantage is the process of clustering similar path will be made easier. Levenshtein Distance is very computational intensive. If there are 2 paths, where Path S has M points and Path T has N Points then the number of calculation that needs to be done is M x N times. By using the new and improved method, although both path may not have the same amount of points, a median path can still be obtain by getting the median points of the two nearest point.

After the Human Detection Function is executed, a Tracking Point (TP) is passed into the Path Detection and Clustering Module. This Tracking Point (TP) is the input of this and through this Tracking Point (TP) we can clearly draw a path that is taken by the monitored individual. In order to obtain the Tracking Point (TP), a simple calculation is done. The calculation is shown as follow:-

Coordinate X = Median of Width of Bounding Box
Coordinate Y = Height of Bounding Box * 0.9

Figure 1 shows the breakdown of the process that occurred in the Path Detection and Clustering Module. Once the Tracking Point (TP) is obtained from the Human Detection Function, a function will be triggered so that a line can be drawn to indicate which path is being taken by the individual that is currently being tracked. This line will be defined as the path that is taken by the individual. This function is being triggered when an individual appears in the video and when the individual goes out from the video, the line that is drawn will be stored into the database and will later on be retrieved to be clustered together with other similar trajectories.

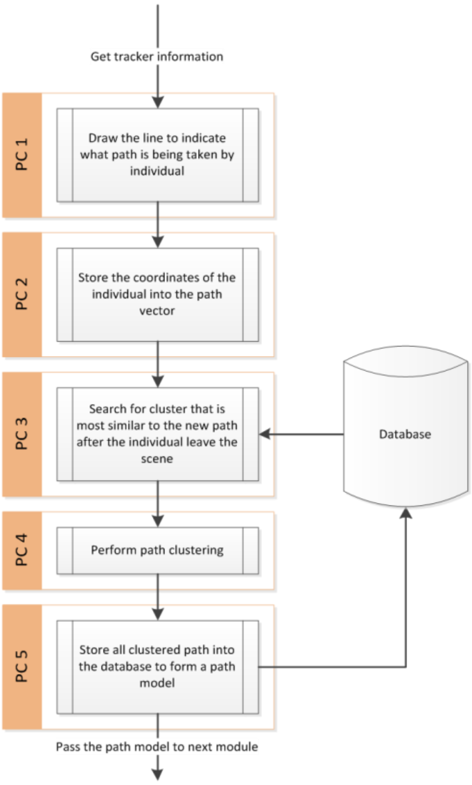

Get tracker information
PC 1
Draw the line to indicate what path is being taken by individual
PC 2
Store the coordinates of the individual into the path vector
PC 3
Search for cluster that is most similar to the new path after the individual leave the scene
Database
PC 4
Perform path clustering
PC 5
Store all clustered path into the database to form a path model
Pass the path model to next module

Figure 1 Path Detection and Clustering Module

	(X_1,Y_1)	...	(X_{1+a},Y_{1+a})	...	(X_{1+b},Y_{1+b})	...	(X_m,Y_m)
(X_1,Y_1)	O	...		...		...	
. . .	. . .	...	. . .	...	. . .	...	. . .
(X_{1+c},Y_{1+c})	X	...	O	...		...	
. . .	. . .	...	. . .	...	. . .	...	. . .
(X_{1+d},Y_{1+d})	X	...	X	...	O	...	
. . .	. . .	...	. . .	...	. . .	...	. . .
(X_n,Y_n)	X	...	X	...	X	...	O

Figure 2 Calculation of Dynamic Time Warping

There are 2 steps in the Dynamic Time Warping (DTW) technique. The first step is to compare each point in one path with every point in the second path with the second path's starting point begins with wherever the previous best point is found, and by doing so a matrix will be generated. The second step is to work through this matrix, starting at the top-left corner (corresponding to the beginning of both paths), and ending at the bottom-right (the end of both paths): for each cell, the Euclidean Distance is calculated and also ignoring the points from the previous calculation as shown in Figure 2. In Figure 2, the "X" means that that particular point does not need to be calculated and could be ignored while the "O" means that two particular points are the closest to each other.

The method that is used to compute the distance between 2 coordinates is the improved version of Levenshtein Distance. The new path clustering method takes 2 paths, S of length M, and T of length N, and returns the clustered oath between the 2 paths. The 2 paths that is used will be represented with 2 array; s[m] and t[n] where m is the number of points that makes up the path s and n is the number of points that makes up the path t. During the beginning of the process of calculation, the distance for the matrix that is used for clustering calculation will be set to maximum value as shown in Figure 2 by using the formula shown in (1). Once the initial value is set, the calculation can proceed to the next step. Starting from the top-left corner of the matrix, the mth coordinate and the nth coordinate will be compared with each other by calculating the distance between the 2 said coordinates by using the formula shown in (2).

$$d(p,q) = \max(s[1 \dots m], t[1 \dots n]) \dots\dots\dots\dots\dots\dots\dots\dots \text{(1)}$$

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \dots\dots \text{(2)}$$

Once the distance between the 2 coordinates have been computed and the value will be stored and once again the matrix for comparing the 2 coordinates will be consulted. After the smallest value have been found in each row j, the next row will begin it's calculation from the column I where i is the coordinate of the previous smallest value of row j. The process will be repeated until the bottom-right cell is reached. When the 2 points that are closest to each other is determined, they will be clustered together by using the formula shown in (3).

$$\mathit{Mean\ Path} = \left(\frac{\mathit{Clustered\ \ x+New\ x}}{2}, \frac{\mathit{Clustered\ \ y+New\ y}}{2}\right) \ldots\ldots\ldots\ldots\ldots \text{(3)}$$

Clustered = coordinates stored in the clustered path

New = coordinates stored in the inserted path for clustering

The new coordinates that is saved in the Mean Path is the Path Model that is normal. A normal path means that the path that is taken is not suspicious as it is frequently being taken by people.

I. Experimental Results

In order to verify the amount of reduced computation during the clustering process, different video sequences were tested. All algorithms are implemented by using a video image size of 320x240 pixels. The image shown below is the example of one of the video scene that is used to test the algorithm.

Figure 4a Background

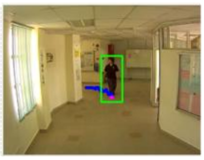


Figure 4b Path Detected

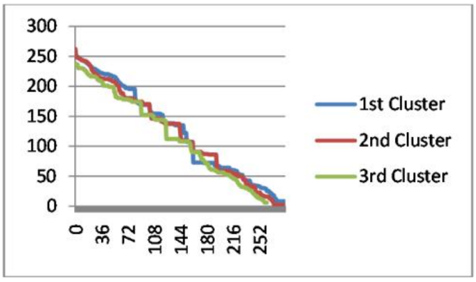


Figure 5 Computational Graph

The Y-axis denotes the number of computation that is required in order to locate the nearest point for clustering while the X-axis denotes the point's position in the main path for clustering. As shown in Figure 5, the number of computation required is decreasing while searching for the closest point for clustering.

II. Conclusions And Future Works

This paper has demonstrated the computation reduction for the path clustering process that is required to generate a Path Model that will determine which path is more frequently taken by passerby that is captured in the scene. Thus, the next step in this research will be to further reduce the computation required by filtering out cluster that is not required for clustering.

References

[1] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, "A Context Based Approach for Detecting Suspicious Behaviors", IEEE TENCON, 2009.

[2] Zhang Zhang, Kaiqi Huang, Tieniu Tan, "Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes", IEEE 2006.

[3] Dimitrios Makris and Tim Ellis, "Path Detection in Video Surveillance", Image and Vision Computing Journal, vol.20/12, pp 895-903, October 2002.

[4] M. Vlachos, G. Kollios, D. Gunopulos, "Discovering Similar Multidimensional Trajectories", Proc. 18th Intl. Conf. on Data Engineering (ICDE'02), pp.673–685, 2002 .

[5] D.Xie, W.Hu, T.Tan, J.Peng, "A Multi-Object Tracking System for Surveillance Video Analysis", in Intl Conf. Pattern Recognition (ICPR'04), 2004 .

[6] F.I.Bashir, A.A.Khokhar, D.Schonfeld, "Segmented trajectory based indexing and retrieval of video data", in Proc. Intl. Conf. on Image Processing (ICIP'03), vol 2, pp.623-626, 2003.

[7] I.Junejo, O.Javed, M. Shah, "Multi Feature Path Modeling for Video Surveillance", in Proc. 17th Intl. Conf. on Pattern Recognition (ICPR'04), vol 2, pp 716-719, 2004.

[8] F.M.Porikli, T.Haga, "Event Detection by Eigenvector Decomposition Using Object and Frame Features", in Proc. Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW'04), vol 7, pp 114-121, 2004.

[9] Z.Fu, W.Hu, T.Tan, "Similarity Based Vehicle Trajectory Clustering and Anomaly Detection", in Proc. Intl. Conf. on Image Processing (ICIP'05), vol 2, pp 602-605, 2005.

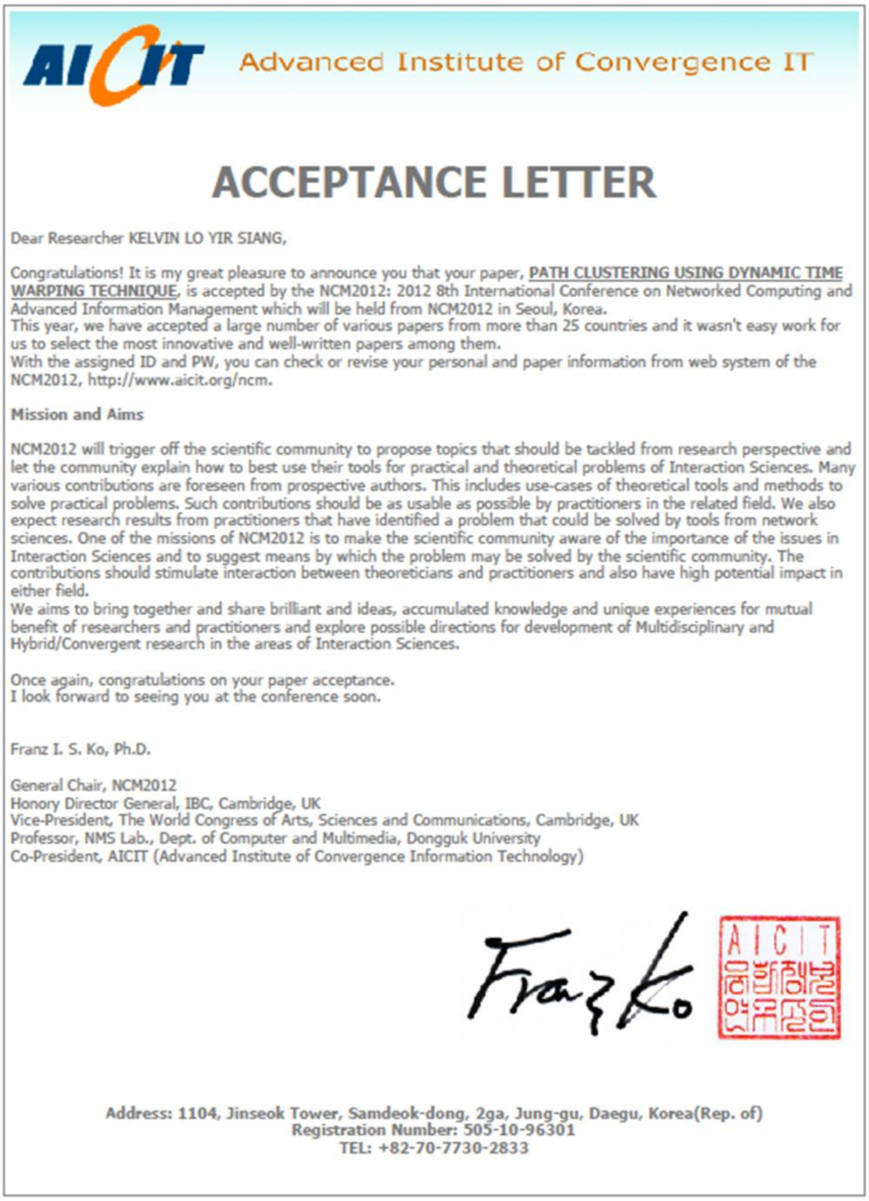
[10] Ujjwal Das Gupta, Vinary Menon and Uday Babbar, "Detecting the Number of Clusters During Expectation-Maximization Clustering Using Information Criterion", IEEE 2010.

[11] Neil Johnson and David Hogg, "Learning the Distribution of Object Trajectories for Event Recognition", British Machine Vision Conference, 2000

[12] E.J.Keogh, M.J.Pazzani, "Scaling up Dynamic Time Warping for Datamining Application", in Proc. sixth ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining (KDD'00), pp285-289, 2000

Appendix B

PUBLICATION – LETTER OF ACCEPTANCE FOR MANUSCRIPT IN Appendix A



AICIT Advanced Institute of Convergence IT

ACCEPTANCE LETTER

Dear Researcher KELVIN LO YIR SIANG,

Congratulations! It is my great pleasure to announce you that your paper, PATH CLUSTERING USING DYNAMIC TIME WARPING TECHNIQUE, is accepted by the NCM2012: 2012 8th International Conference on Networked Computing and Advanced Information Management which will be held from NCM2012 in Seoul, Korea.
This year, we have accepted a large number of various papers from more than 25 countries and it wasn't easy work for us to select the most innovative and well-written papers among them.
With the assigned ID and PW, you can check or revise your personal and paper information from web system of the NCM2012, http://www.aicit.org/ncm.

Mission and Aims

NCM2012 will trigger off the scientific community to propose topics that should be tackled from research perspective and let the community explain how to best use their tools for practical and theoretical problems of Interaction Sciences. Many various contributions are foreseen from prospective authors. This includes use-cases of theoretical tools and methods to solve practical problems. Such contributions should be as usable as possible by practitioners in the related field. We also expect research results from practitioners that have identified a problem that could be solved by tools from network sciences. One of the missions of NCM2012 is to make the scientific community aware of the importance of the issues in Interaction Sciences and to suggest means by which the problem may be solved by the scientific community. The contributions should stimulate interaction between theoreticians and practitioners and also have high potential impact in either field.
We aims to bring together and share brilliant and ideas, accumulated knowledge and unique experiences for mutual benefit of researchers and practitioners and explore possible directions for development of Multidisciplinary and Hybrid/Convergent research in the areas of Interaction Sciences.

Once again, congratulations on your paper acceptance.
I look forward to seeing you at the conference soon.

Franz I. S. Ko, Ph.D.

General Chair, NCM2012
Honory Director General, IBC, Cambridge, UK
Vice-President, The World Congress of Arts, Sciences and Communications, Cambridge, UK
Professor, NMS Lab., Dept. of Computer and Multimedia, Dongguk University
Co-President, AICIT (Advanced Institute of Convergence Information Technology)

Franz Ko

AICIT
융합정보
연구학회

Address: 1104, Jinseok Tower, Samdeok-dong, 2ga, Jung-gu, Daegu, Korea(Rep. of)
Registration Number: 505-10-96301
TEL: +82-70-7730-2833

Appendix C

PUBLICATION (International Journal of Information Processing and Management) – PATH CLUSTERING USING A MIXTURE OF DYNAMIC TIME WARPING TECHNIQUE

Path Clustering using a Mixture of Dynamic Time Warping Technique

Kelvin Lo Yir Siang, Siak Wang Khor
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman
Kuala Lumpur, Malaysia
E-mail: ken_lo74@hotmail.com, khorsw@utar.edu.my

Abstract

In order to monitor the safety and security of an area, video surveillance system is deployed and implemented in the said area. Such video surveillance system usually relies on the detection of suspicious behavior that is captured by the surveillance camera. In this paper, we present a novel method for clustering similar trajectories in video surveillance system. The purpose of performing trajectories clustering is to build a path model which can be used to detect any suspicious activity in the monitored scene. Path models are learnt from the accumulation of trajectory data over long time periods, and can be used to augment the classification of subsequent track data. This approach does not employ the traditional way of constructing path model, yet it simplifies the computation in the process of clustering similar trajectories by calculating average path for each detected and matched trajectory. The results demonstrate the efficiency of the proposed approach in clustering the path for detecting deviant walking paths.

Keywords: *path clustering, dynamic time warping, path model, levenshtein distance, video surveillance*

1. Introduction

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform CCTV systems from 'post-mortem tools' into the forefront of crime-fighting tools. The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel [1].

One of the factors is that human operators have fatigue limits when observing several monitoring screens for prolonged periods of time. When this is reached monitoring performance on surveillance areas will be degraded. Another factor is that there is a limitation in the number of monitors that a human operator can manage at the same time. This would add more complexity in large-scale surveillance systems having many cameras with a limited number of security staff. Therefore, an automatic process detecting suspicious behavior is a necessity. In order to devise such an automatic process, there are many issues that need to be solved and one of them is the creation of path model for the automated system.

In path modeling and surveillance, the objective here is to build a system that, once given a large amount of set of acceptable trajectories (e.g. normal path) of objects in a scene, is able to build a path model. This is to fulfill the aim to learn the routes or paths most commonly taken by objects as they traverse through a scene. Once the model for the scene is created, the method should be able to classify incoming trajectories as conforming to the said model or not. Moreover, as common pathways are detected by clustering the trajectories, assigning detected trajectory its associated path model can be done efficiently thereby only storing the path label and the object labels instead of the whole trajectory set, resulting in a significant compression for storing surveillance data.[2] For similarity based clustering, a key issue is how to measure the similarity between two trajectories [3].

The following section gives a brief overview of some of the previous works done by other researchers on the path detection and clustering domain and the validation of the path taken. Section 3 describes the process of trajectories clustering in building the path model for the surveillance system and also how to determine the validity of the path taken. Then, the results for applying the averaging trajectories clustering method on image sequences captured are presented. And lastly, conclusions are drawn.

2. Related Works

There are many approaches that have been research and taken to cluster all the similar paths in order to create a path model and one of the methods is using a spatial model to represent the routes in an image which is used by Dimitrios Makris et al [4]. A trajectory is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function; otherwise a new route is created for the new trajectory. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person.

M.Vlachos et al. introduces a method called The Longest Common Subsequence (LCSS) distance [5]. This method finds the alignment between two sequences that maximize the length of common subsequence. Although this method is classified as one of the more efficient method to cluster similar trajectory, it is however geared more towards recognizing shape similarity rather than path trajectory [6].

F.I.Bashir et al. proposed a method called the Principle Components Analysis (PCA) where a trajectory is first represented as a 1-D signal by concatenating the x- and the y-projections [7]. Then the signal is converted into the first few PCA coefficients. The trajectory similarity is computed as the Euclidean distance between the PCA coefficients. There is a disadvantage to this method. It is brittle to the distortion and other noise. However, it can cluster similar trajectory in a short amount of time.

Hausdorff Distance is a method that is used by I.Junejo et al. to cluster similar trajectory [8]. The method get the longest distance among the two shortest distances between two coordinates. The advantage of this method is that it can get the shortest distance after comparing with all possible possibilities. However, this method is unable to properly recognize the direction of the trajectory. The same path but different direction is the same before this method.

F.M. Porikli et al. mentioned that using Hidden Markov Model-Based Distance is a viable solution to cluster similar trajectory [9]. By comparing the likelihood of the trajectories to their own model, similar trajectory will be clustered together. By using probabilities, it is indeed an efficient method. However, due to limited number of training data, it will most likely suffer from over-fitting.

Another approach of clustering similar trajectory would be using the Euclidean Distance Method that is used by Z.Fu et al [10]. By getting all the coordinates of the similar trajectory, a simple calculation will be computed and the median will be plotted out. The advantage of using this method lies with the simplicity of the calculation. However, because the calculation is simple, it is also computational extensive and thus it is also a disadvantage of using this method.

One clustering approach that is implemented by Ujjwal Das Guptaet al. is to assume a Gaussian Mixture Model and implementing the X-Means algorithm [11]. In this model, the input data is assumed to have been generated by selecting any one of the K Gaussian distributions. Each cluster is thus represented by a single distribution. The major problem that is faced by using the X-Means is that is assumes an identical spherical Gaussian of the data and thus it tends to overfit data in elliptical clusters. The advantages of using this algorithm is that can optimally detect the clusters for a chosen value of K. The method is further improved again by R.Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani and M. Ebrahimi in the journal they wrote in 2010. [12]

Neil Johnson and David Hogg present a statistically based model of object trajectories which is learnt from image sequences as a viable solution to cluster similar trajectory and create a path model [13]. Trajectory data is supplied by a tracker using Active Shape Models, from which a model of the distribution of typical trajectories is learnt. From the sequence of the image shown during the training stage, the trajectories of the tracked individual can be obtained. However, because the amount of data is too large, a large storage is needed.

Once a trajectory of a moving object is obtained, it is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function. One limitation of this approach is that only spatial information is used for trajectory clustering. However, Barak Raveh, Angela Enosh and Dan Halperin introduced a simple algorithm to merge an arbitrary number of input motion paths into a hybrid output path of superior quality. [14] Their approach is based on the observation that the quality of certain subpaths within each solution may be higher thatn the quality of the entire path and thus the algorithm.

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform Closed Circuit Television (CCTV) systems from ‘post-mortem tools’ into the forefront of crime-fighting tools. The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel.

Hidden Markov Model is also another method to detect suspicious behavior. Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato and Yozo Fujino propose a method detects anomalies and classifies into several behavior patterns based on probabilistic distances between HMMs. [15] They have tested the method using an online dataset and also implement unsupervised training. The advantages of using this method is that the system that implements this method can be used anywhere without any prior knowledge of the area.

Another method would was proposed by a paper that is written by Simon Fong and Zhuang Yan. Their methods are referring to user activities and actions that are reflected by their movement [16] They argue that by connecting multiple access reference points, they can gain a better understanding of the user’s behavior than a single entry validation verdict such as access “Granted” or “Denied”. Their solution adopted SRI’s NIDES technique, for implementing a user’s profile that has a set of statistical measures. The advantage of using this method is that the user’s patterns can be learnt and the system is able to tell whether a new trail is normal or suspicious.

Another way to detect suspicious behavior would be the Mixture of Von Mises method that was proposed by Simone Calderara, Rita Cucchiara and Andrea Prati. They decided to model the behavior in terms of the trajectory of movement, modeled as a sequence of directions computed as the angle between two consecutive positions and that allows modeling robustly trajectories even in presence of noisy observations. [17] The disadvantage of this method is that it would be hard to create a training set. On the other hand, this method is both intuitive and discriminative enough to detect suspicious behavior efficiently. In another work of the same author, they made an improvement. By including the Expectation Maximization, they can now implement the method using an unsupervised learning. The previous disadvantage of no training set is available is now solved. [18]

Daniel Barbara and Maurizio Filippone have proposed a design of a novel technique that will detect suspicious behavior. Their implementation uses standard techniques for foreground and background object detection in images. [19] First, they will take an image of a foreground. Next, any object that is not present in the foreground and yet has appeared on video is considered suspicious. This method is useful in detecting suspicious behavior or suspicious object. However, the downside of this method is that they must first define a meaningful set of features, and a proper distance measure between images or else this method is not applicable.

Detecting uncommon trajectories is also a feasible method. This method is proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda. They used a dataset from CAVIAR 1 which are available publicly. For the purpose of the experiment, they only stick to the basic lower level processing and focus more on the tracking and detection of suspicious behavior. They will first trace the people’s position in each image sequence and the person’s trajectory will be formed up from these coordinates. Then, from the trajectory, they will then group them into clusters of similar trajectories. Those trajectories that do not fall into any of the clusters will be deemed suspicious. [20] The disadvantages of using this heuristic method is that it will not be very effective when dealing with the real life situation as all the suspicious behavior pattern that is used is mainly created for experiment purposes.

The solution that were proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda explore two contextual features, such as type of behavior as well as the commonality index of each behavior type as contextual information, to detect suspicious behavior. This solution uses shape feature to calculate the common shape of walking paths in airport scenario. Any walking path deviated much from the common shape is labeled as suspicious. [21] The advantage of this method is that it will be more adaptable to the unseen patterns and it does not need the training sets to include all possible types of normal behaviors. However, this model will only work well provided that the database for all kind of possible attack is available. Thus, newly created attacks will not be detected. Yi Feng Zhou and Jim P.Y. Lee however show that this problem can be solved. [22] The method proposed by them is a modified expectation maximization algorithm for maximum likelihood direction of arrival estimation can be used for real life update and thus newly created attacks that cannot be detected by Arnold and his partner’s work but can be detected by Yi Feng Zhou and Jim P.Y. lee’s work.

Xiaogang Wang, Kinh Tieu, and W. Eric L. Grimson however proposed a novel approach for activity analysis in multiple synchronized but uncalibrated static camera views. [23] They refer to activities as motion patterns of objects, which correspond to paths in far-field scenes. They assume that the topology of cameras is unknown and quite arbitrary, the fields of views covered by these cameras may have no overlap or any amount of independently, and the positions and velocitied of objects along trajectories are computed as features.

3. Proposed Methodology

In order to determine the individual's status based on "Deviation from Normal Path" factor, a normal path model that is created through training is needed. The normal path model will be applied to the system as based and any path that is taken by the monitored individual that is deviated from the normal path that is already defined in the normal path model will be deemed as suspicious.

Clustering of the similar trajectory is being done by using Dynamic Time Warping (DTW). The DTW algorithm uses a dynamic programming technique to solve this problem. The reason this method is being used is because it is not computational intensive and it can cluster similar trajectory at a very short amount of time. Furthermore, should 2 path with different numbers of points will still be able to be clustered together to form a normal path. The basic requirement for Dynamic Time Warping is the coordinates that is obtained from both the clustered path and also the newly obtained path. We will calculate the distance of each coordinates from both path and each coordinates will be saved in an array and thus 2 arrays are obtained.

We will be using an improved version of Levenshtein Distance to compare both coordinates. Using this distance has 2 advantages. The first advantage will be both path does not need to have the same amount of points to be compared and the second advantage is the process of clustering similar path will be made easier. Levenshtein Distance is very computational intensive. If there are 2 paths, where Path S has M points and Path T has N Points, then the number of calculation that needs to be done is M x N times. By using the new and improved method, although both path may not have the same amount of points, a median path can still be obtain by getting the median points of the two nearest point.

After the Human Detection Function is executed, a Tracking Point (TP) is passed into the Path Detection and Clustering Module. This Tracking Point (TP) is the input of this and through this Tracking Point (TP) we can clearly draw a path that is taken by the monitored individual. In order to obtain the Tracking Point (TP), a simple calculation is done. The calculation is shown as follow:-

Coordinate X = Median of Width of Bounding Box

Coordinate Y = Height of Bounding Box * 0.9

Figure 1 shows the breakdown of the process that occurred in the Path Detection and Clustering Module. Once the Tracking Point (TP) is obtained from the Human Detection Function, a function will be triggered so that a line can be drawn to indicate which path is being taken by the individual that is currently being tracked. This line will be defined as the path that is taken by the individual. This function is being triggered when an individual appears in the video and when the individual goes out from the video, the line that is drawn will be stored into the database and will later on be retrieved to be clustered together with other similar trajectories.

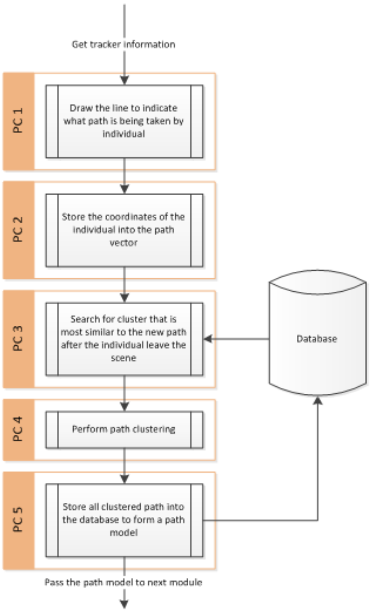
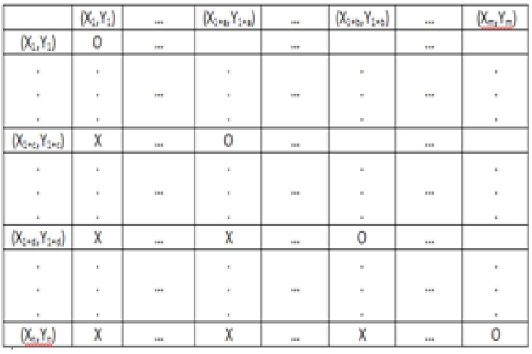

Get tracker information
PC 1
Draw the line to indicate what path is being taken by individual
PC 2
Store the coordinates of the individual into the path vector
PC 3
Search for cluster that is most similar to the new path after the individual leave the scene
Database
PC 4
Perform path clustering
PC 5
Store all clustered path into the database to form a path model
Pass the path model to next module

Figure 1 Path Detection and Clustering Module

Firstly, the first point of the path that is just retrieved will be compared with the entire path in the cluster that is stored in the database in order to only obtain the path that is has similar first point with the retrieved path. By doing so, there is no need to transverse and compare the entire path in the database in order to perform path comparison and clustering. By doing so, the number of processes and computation that is needed to be done will be reduced. Once the similar path have been retrieved from the database, path clustering using Dynamic Time Warping technique will commence.



	(X_1,Y_1)	...	(X_{1+a},Y_{1+a})	...	(X_{1+b},Y_{1+b})	...	(X_m,Y_m)
(X_1,Y_1)	O	...		...		...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_{1+c},Y_{1+c})	X	...	O	...		...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_{1+d},Y_{1+d})	X	...	X	...	O	...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_n,Y_n)	X	...	X	...	X	...	O

Figure 2 Calculation of Dynamic Time Warping

There are 2 steps in the Dynamic Time Warping (DTW) technique. The first step is to compare each point in one path with every point in the second path with the second path's starting point begins with wherever the previous best point is found, and by doing so a matrix will be generated. The second step is to work through this matrix, starting at the top-left corner (corresponding to the beginning of both paths), and ending at the bottom-right (the end of both paths): for each cell, the Euclidean Distance is calculated and also ignoring the points from the previous calculation as shown in Figure 2. In Figure 2, the "X" means that that particular point does not need to be calculated and could be ignored while the "O" means that two particular points are the closest to each other.

The method that is used to compute the distance between 2 coordinates is the improved version of Levenshtein Distance. The new path clustering method takes 2 paths, S of length M, and T of length N, and returns the clustered oath between the 2 paths. The 2 paths that is used will be represented with 2 array; s[m] and t[n] where m is the number of points that makes up the path s and n is the number of points that makes up the path t. During the beginning of the process of calculation, the distance for the matrix that is used for clustering calculation will be set to maximum value as shown in Figure 2 by using the formula shown in (1). Once the initial value is set, the calculation can proceed to the next step. Starting from the top-left corner of the matrix, the mth coordinate and the nth coordinate will be compared with each other by calculating the distance between the 2 said coordinates by using the formula shown in (2).

$$d(p,q) = \max(s[1 \ldots m], t[1 \ldots n]) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \ldots\ldots \text{(2)}$$

Once the distance between the 2 coordinates have been computed and the value will be stored and once again the matrix for comparing the 2 coordinates will be consulted. After the smallest value have been found in each row j, the next row will begin it's calculation from the column I where i is the coordinate of the previous smallest value of row j. The process will be repeated until the bottom-right cell is reached. When the 2 points that are closest to each other is determined, they will be clustered together by using the formula shown in (3).

$$Mean\ Path = \left(\frac{Clustered\ \ x + New\ x}{2}, \frac{Clustered\ \ y + New\ y}{2}\right) \ldots\ldots\ldots\ldots\ldots \text{(3)}$$

Clustered = coordinates stored in the clustered path

New = coordinates stored in the inserted path for clustering

The new coordinates that is saved in the Mean Path is the Path Model that is normal. A normal path means that the path that is taken is not suspicious as it is frequently being taken by people.

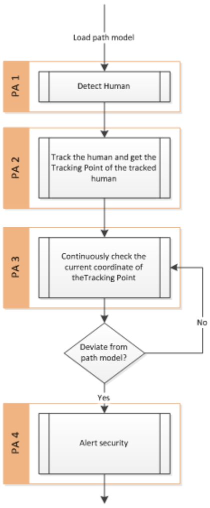

Load path model
PA 1
Detect Human
PA 2
Track the human and get the Tracking Point of the tracked human
PA 3
Continuously check the current coordinate of theTracking Point
No
Deviate from path model?
Yes
PA 4
Alert security

Figure 4 Path Analysis Module

The breakdown of the Path Analysis Module is shown in Figure 4.The Path Analysis Module will determine the status of the currently monitored individual based on deviation from Path Model. In order to determine the individual's status based on "deviation from Path Model" factor, a normal path model that is created through training by Module 2 is needed. The normal path model will be applied to the system as a reference guide and any path that is taken by the monitored individual deviated from the normal will be deemed as suspicious. When the individual is being monitored, the TP is being stored into the database as well.

With the coordinates, a path will be drawn and that path will constantly be compared with the current normal path model's clustered path. If the path that is taken by the monitored individual is still within the Region of Interest (ROI) that is created by thresholding the clustered normal path, then the individual is not suspicious. Otherwise, security personal will be alerted. Figure 5 shows the example of the thresholded clustered path.

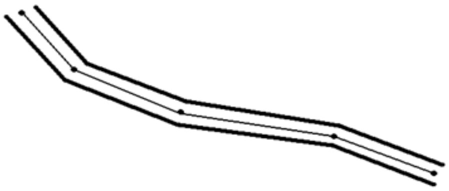
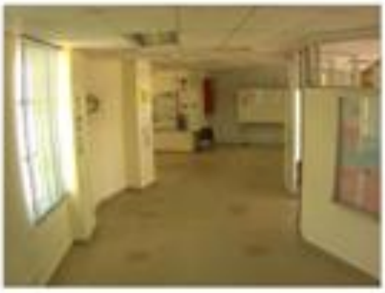
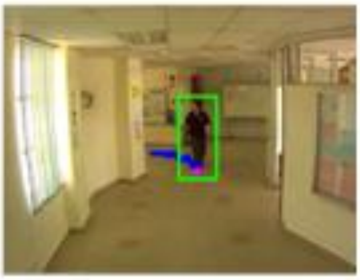


Figure 5 Thresholded Clustered Path

4.Experimental Results

In order to verify the amount of reduced computation during the clustering process, different video sequences were tested. All algorithms are implemented by using a video image size of 320x240 pixels. The image shown below is the example of one of the video scene that is used to test the algorithm.





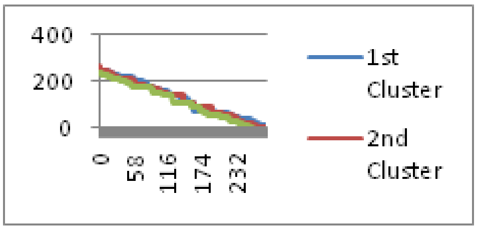


Figure 6a Background

Figure 6b Path Detected

Figure 7 Computation Graph

The Y-axis denotes the number of computation that is required in order to locate the nearest point for clustering while the X-axis denotes the point's position in the main path for clustering. As shown in Figure 7, the number of computation required is decreasing while searching for the closest point for clustering. By using the cluster index selection method together with the DTW technique, the overall performance of the system has improved.

5. Conclusions and Future Works

This paper has demonstrated the computation reduction for the path clustering process that is required to generate a Path Model that will determine which path is more frequently taken by passerby that is captured in the scene. Thus, the next step in this research will be to further enhance the accuracy of the system with better path retrieval method and further reduce the number of computation needed.

6. References

[1] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “A Context Based Approach for Detecting Suspicious Behaviors”, IEEE TENCON, 2010.

[2] Kenneth Rose, Member, IEEE, Eitan Gurewitz, and Geoffrey C. Fox “Constrained Clustering as an Optimization Method”, IEEE Journal and Magazine 1993.

[3] Zhang Zhang, Kaiqi Huang, Tieniu Tan, “Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes”, IEEE 2006.

[4] Dimitrios Makris and Tim Ellis, "Path Detection in Video Surveillance", Image and Vision Computing Journal, vol.20/12, pp 895-903, October 2002.

[5] M. Vlachos, G. Kollios, D. Gunopulos, “Discovering Similar Multidimensional Trajectories”, Proc. 18th Intl. Conf. on Data Engineering (ICDE’02), pp.673–685, 2002 .

[6] D.Xie, W.Hu, T.Tan, J.Peng, “A Multi-Object Tracking System for Surveillance Video Analysis”, in Intl Conf. Pattern Recognition (ICPR’04), 2004 .

[7] F.I.Bashir, A.A.Khokhar, D.Schonfeld, “Segmented trajectory based indexing and retrieval of video data”, in Proc. Intl. Conf. on Image Processing (ICIP’03), vol 2, pp.623-626, 2003.

[8] I.Junejo, O.Javed, M. Shah, “Multi Feature Path Modeling for Video Surveillance”, in Proc. 17th Intl. Conf. on Pattern Recognition (ICPR’04), vol 2, pp 716-719, 2004.

[9] F.M.Porikli, T.Haga, “Event Detection by Eigenvector Decomposition Using Object and Frame Features”, in Proc. Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW’04), vol 7, pp 114-121, 2004.

[10] Z.Fu, W.Hu, T.Tan, “Similarity Based Vehicle Trajectory Clustering and Anomaly Detection”, in Proc. Intl. Conf. on Image Processing (ICIP’05), vol 2, pp 602-605, 2005.

[11] Ujjwal Das Gupta, Vinary Menon and Uday Babbar, “Detecting the Number of Clusters During Expectation-Maximization Clustering Using Information Criterion”, IEEE 2010.

[12] R.Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani and M. Ebrahimi, “Vanishing point detection in corridors: using Hough transform and K-means clustering”, IEEE Journal and Magazine 2010.

[13] Neil Johnson and David Hogg, “Learning the Distribution of Object Trajectories for Event Recognition”, British Machine Vision Conference, 2000

[14] Barak Raveh, Angela Enosh, and Dan Halperin, “A Little More, a Lot Better: Improving Path Quality by a Path-Merging Algorithm”, IEEE Journal and Magazine 2011.

[15] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato and Yozo Fujino, “Learning Motion Patterns and Anomaly Detection by Human Trajectory analysis”, IEEE 2007.

[16] Simon Fong and Zhuang Yan, “A security Model For Detecting Suspicious Patterns in Physical Environment”, IEEE, 2007

[17] Simone Calderara, Rita Cucchiara and Andrea Prati, “Detection of Abnormal Behaviors Using A Mixture of Von Mises Distributions”, IEEE 2007.

[18] Simone Calderara, Rita Cucchiara and Andrea Prati, “Using Circular Statistics for Trajectory Shape Analysis”, IEEE 2008.

[19] Daniel Barbara and Maurizio Filippone, “Detecting Suspicious Behaviour in Surveilance Images”, IEEE 2008.

[20] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “Detecting Uncommon Trajectories”, IEEE, 2008

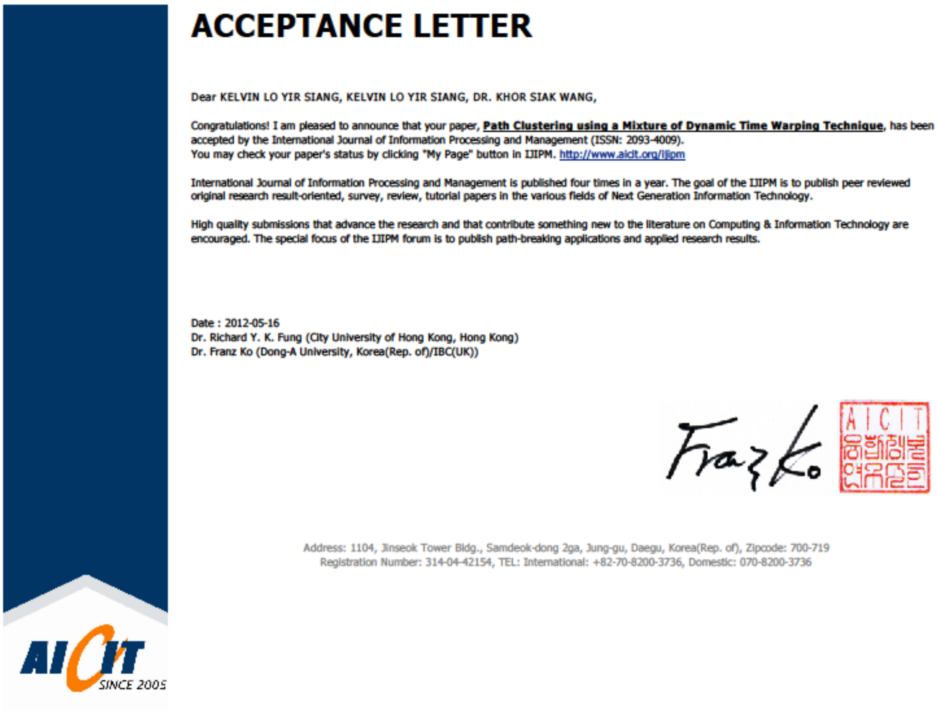
[21] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “A Context Based Approach for Detecting Suspicious Behaviors”, IEEE TENCON, 2009.

[22] Yi Feng Zhou and Jim P.Y. Lee, “A Modified Expectation Maximization Algorithm for Maximum Likelihood Direction of arrival Estimation”, IEEE 2010.

[23] Xiaogang Wang, Kinh Tieu, and W. Eric L. Grimson, “Correspondence-Free Activity Analysis and Scene Modeling in Multiple Camera Views”, IEEE Journal and Magazine 2010.

Appendix D

PUBLICATION – LETTER OF ACCEPTANCE FOR MANUSCRIPT IN Appendix C



ACCEPTANCE LETTER

Dear KELVIN LO YIR SIANG, KELVIN LO YIR SIANG, DR. KHOR SIAK WANG,

Congratulations! I am pleased to announce that your paper, **Path Clustering using a Mixture of Dynamic Time Warping Technique**, has been accepted by the International Journal of Information Processing and Management (ISSN: 2093-4009).
You may check your paper's status by clicking "My Page" button in IJIPM. http://www.aicit.org/ijipm

International Journal of Information Processing and Management is published four times in a year. The goal of the IJIPM is to publish peer reviewed original research result-oriented, survey, review, tutorial papers in the various fields of Next Generation Information Technology.

High quality submissions that advance the research and that contribute something new to the literature on Computing & Information Technology are encouraged. The special focus of the IJIPM forum is to publish path-breaking applications and applied research results.

Date : 2012-05-16
Dr. Richard Y. K. Fung (City University of Hong Kong, Hong Kong)
Dr. Franz Ko (Dong-A University, Korea(Rep. of)/IBC(UK))

Franz Ko

AICIT

Address: 1104, Jinseok Tower Bldg., Samdeok-dong 2ga, Jung-gu, Daegu, Korea(Rep. of), Zipcode: 700-719
Registration Number: 314-04-42154, TEL: International: +82-70-8200-3736, Domestic: 070-8200-3736

AICIT SINCE 2005

Appendix E

PUBLICATION (International Journal of Intelligent Information Processing) – PATH CLUSTERING USING A MIXTURE OF DYNAMIC TIME WARPING TECHNIQUE AND CLUSTER SEARCHING INDEX METHOD

Path Clustering using a Mixture of Dynamic Time Warping Technique and Cluster Searching Index Method

Kelvin Lo Yir Siang, Siak Wang Khor
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman
Kuala Lumpur, Malaysia
E-mail: ken_lo74@hotmail.com, khorsw@utar.edu.my

Abstract

In order to monitor the safety and security of an area, video surveillance system is deployed and implemented in the said area. Such video surveillance system usually relies on the detection of suspicious behavior that is captured by the surveillance camera. In this paper, we present a novel method for clustering similar trajectories in video surveillance system. The purpose of performing trajectories clustering is to build a path model which can be used to detect any suspicious activity in the monitored scene. Path models are learnt from the accumulation of trajectory data over long time periods, and can be used to augment the classification of subsequent track data. This approach does not employ the traditional way of constructing path model, yet it simplifies the computation in the process of clustering similar trajectories by calculating average path for each detected and matched trajectory. The results demonstrate the efficiency of the proposed approach in clustering the path for detecting deviant walking paths.

Keywords: *path clustering, dynamic time warping, path model, levenshtein distance, video surveillance*

1. Introduction

One of the factors is that human operators have fatigue limits when observing several monitoring screens for prolonged periods of time. When this is reached monitoring performance on surveillance areas will be degraded. Another factor is that there is a limitation in the number of monitors that a human operator can manage at the same time. This would add more complexity in large-scale surveillance systems having many cameras with a limited number of security staff. Therefore, an automatic process detecting suspicious behavior is a necessity. In order to devise such an automatic process, there are many issues that need to be solved and one of them is the creation of path model for the automated system.

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform CCTV systems from ‘post-mortem tools’ into the forefront of crime-fighting tools. The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel [1].

In path modeling and surveillance, the objective here is to build a system that, once given a large amount of set of acceptable trajectories (e.g. normal path) of objects in a scene, is able to build a path model. This is to fulfill the aim to learn the routes or paths most commonly taken by objects as they traverse through a scene. Once the model for the scene is created, the method should be able to classify incoming trajectories as conforming to the said model or not. Moreover, as common pathways are detected by clustering the trajectories, assigning detected trajectory its associated path model can be done efficiently thereby only storing the path label and the object labels instead of the whole trajectory set, resulting in a significant compression for storing surveillance data.[2] For similarity based clustering, a key issue is how to measure the similarity between two trajectories [3].

The following section gives a brief overview of some of the previous works done by other researchers on the path detection and clustering domain and the validation of the path taken. Section 3 describes the process of trajectories clustering in building the path model for the surveillance system and also how to determine the validity of the path taken. Then, the results for applying the averaging trajectories clustering method on image sequences captured are presented. And lastly, conclusions are drawn.

2. Related Works

There are many approaches that have been research and taken to cluster all the similar paths in order to create a path model and one of the methods is using a spatial model to represent the routes in an image which is used by Dimitrios Makris et al [4]. A trajectory is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function; otherwise a new route is created for the new trajectory. One limitation of this approach is that only spatial information is used for trajectory clustering and behavior recognition. The system cannot distinguish between a person walking and a person lingering around, or between a running and a walking person.

M.Vlachos et al. introduces a method called The Longest Common Subsequence (LCSS) distance [5]. This method finds the alignment between two sequences that maximize the length of common subsequence. Although this method is classified as one of the more efficient method to cluster similar trajectory, it is however geared more towards recognizing shape similarity rather than path trajectory [6].

F.I.Bashir et al. proposed a method called the Principle Components Analysis (PCA) where a trajectory is first represented as a 1-D signal by concatenating the x- and the y-projections [7]. Then the signal is converted into the first few PCA coefficients. The trajectory similarity is computed as the Euclidean distance between the PCA coefficients. There is a disadvantage to this method. It is brittle to the distortion and other noise. However, it can cluster similar trajectory in a short amount of time.

Hausdorff Distance is a method that is used by I.Junejo et al. to cluster similar trajectory [8]. The method get the longest distance among the two shortest distances between two coordinates. The advantage of this method is that it can get the shortest distance after comparing with all possible possibilities. However, this method is unable to properly recognize the direction of the trajectory. The same path but different direction is the same before this method.

F.M. Porikli et al. mentioned that using Hidden Markov Model-Based Distance is a viable solution to cluster similar trajectory [9]. By comparing the likelihood of the trajectories to their own model, similar trajectory will be clustered together. By using probabilities, it is indeed an efficient method. However, due to limited number of training data, it will most likely suffer from over-fitting.

Another approach of clustering similar trajectory would be using the Euclidean Distance Method that is used by Z.Fu et al [10]. By getting all the coordinates of the similar trajectory, a simple calculation will be computed and the median will be plotted out. The advantage of using this method lies with the simplicity of the calculation. However, because the calculation is simple, it is also computational extensive and thus it is also a disadvantage of using this method.

One clustering approach that is implemented by Ujjwal Das Guptaet al. is to assume a Gaussian Mixture Model and implementing the X-Means algorithm [11]. In this model, the input data is assumed to have been generated by selecting any one of the K Gaussian distributions. Each cluster is thus represented by a single distribution. The major problem that is faced by using the X-Means is that is assumes an identical spherical Gaussian of the data and thus it tends to overfit data in elliptical clusters. The advantages of using this algorithm is that can optimally detect the clusters for a chosen value of K. The method is further improved again by R.Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani and M. Ebrahimi in the journal they wrote in 2010. [12]

Neil Johnson and David Hogg present a statistically based model of object trajectories which is learnt from image sequences as a viable solution to cluster similar trajectory and create a path model [13]. Trajectory data is supplied by a tracker using Active Shape Models, from which a model of the distribution of typical trajectories is learnt. From the sequence of the image shown during the training stage, the trajectories of the tracked individual can be obtained. However, because the amount of data is too large, a large storage is needed.

Once a trajectory of a moving object is obtained, it is matched with routes already existing in a database using a simple distance measure. If a match is found, the existing route is updated by a weight update function. One limitation of this approach is that only spatial information is used for trajectory clustering. However, Barak Raveh, Angela Enosh and Dan Halperin introduced a simple algorithm to merge an arbitrary number of input motion paths into a hybrid output path of superior quality. [14] Their approach is based on the observation that the quality of certain subpaths within each solution may be higher thatn the quality of the entire path and thus the algorithm.

Suspicious behavior detection is one of the paramount goals in surveillance systems. However, it is constrained by human factors which have posed major difficulties for changing the utilization of the systems from after-crime-scene tools to the forefront crime-fighting tools. A surveillance system with such a capability can transform Closed Circuit Television (CCTV) systems from ‘post-mortem tools’ into the forefront of crime-fighting tools. The security personnel can act as soon as there is a suspicious activity detected and hence could stop the crime before it is committed. This could also solve the problems of scarcity of security personnel.

Hidden Markov Model is also another method to detect suspicious behavior. Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato and Yozo Fujino propose a method detects anomalies and classifies into several behavior patterns based on probabilistic distances between HMMs. [15] They have tested the method using an online dataset and also implement unsupervised training. The advantages of using this method is that the system that implements this method can be used anywhere without any prior knowledge of the area.

Another method would was proposed by a paper that is written by Simon Fong and Zhuang Yan. Their methods are referring to user activities and actions that are reflected by their movement [16] They argue that by connecting multiple access reference points, they can gain a better understanding of the user’s behavior than a single entry validation verdict such as access “Granted” or “Denied”. Their solution adopted SRI’s NIDES technique, for implementing a user’s profile that has a set of statistical measures. The advantage of using this method is that the user’s patterns can be learnt and the system is able to tell whether a new trail is normal or suspicious.

Another way to detect suspicious behavior would be the Mixture of Von Mises method that was proposed by Simone Calderara, Rita Cucchiara and Andrea Prati. They decided to model the behavior in terms of the trajectory of movement, modeled as a sequence of directions computed as the angle between two consecutive positions and that allows modeling robustly trajectories even in presence of noisy observations. [17] The disadvantage of this method is that it would be hard to create a training set. On the other hand, this method is both intuitive and discriminative enough to detect suspicious behavior efficiently. In another work of the same author, they made an improvement. By including the Expectation Maximization, they can now implement the method using an unsupervised learning. The previous disadvantage of no training set is available is now solved. [18]

Daniel Barbara and Maurizio Filippone have proposed a design of a novel technique that will detect suspicious behavior. Their implementation uses standard techniques for foreground and background object detection in images. [19] First, they will take an image of a foreground. Next, any object that is not present in the foreground and yet has appeared on video is considered suspicious. This method is useful in detecting suspicious behavior or suspicious object. However, the downside of this method is that they must first define a meaningful set of features, and a proper distance measure between images or else this method is not applicable.

Detecting uncommon trajectories is also a feasible method. This method is proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda. They used a dataset from CAVIAR 1 which are available publicly. For the purpose of the experiment, they only stick to the basic lower level processing and focus more on the tracking and detection of suspicious behavior. They will first trace the people’s position in each image sequence and the person’s trajectory will be formed up from these coordinates. Then, from the trajectory, they will then group them into clusters of similar trajectories. Those trajectories that do not fall into any of the clusters will be deemed suspicious. [20] The disadvantages of using this heuristic method is that it will not be very effective when dealing with the real life situation as all the suspicious behavior pattern that is used is mainly created for experiment purposes.

The solution that were proposed by Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda explore two contextual features, such as type of behavior as well as the commonality index of each behavior type as contextual information, to detect suspicious behavior. This solution uses shape feature to calculate the common shape of walking paths in airport scenario. Any walking path deviated much from the common shape is labeled as suspicious. [21] The advantage of this method is that it will be more adaptable to the unseen patterns and it does not need the training sets to include all possible types of normal behaviors. However, this model will only work well provided that the database for all kind of possible attack is available. Thus, newly created attacks will not be detected. Yi Feng Zhou and Jim P.Y. Lee however show that this problem can be solved. [22] The method proposed by them is a modified expectation maximization algorithm for maximum likelihood direction of arrival estimation can be used for real life update and thus newly created attacks that cannot be detected by Arnold and his partner’s work but can be detected by Yi Feng Zhou and Jim P.Y. lee’s work.

Xiaogang Wang, Kinh Tieu, and W. Eric L. Grimson however proposed a novel approach for activity analysis in multiple synchronized but uncalibrated static camera views. [23] They refer to activities as motion patterns of objects, which correspond to paths in far-field scenes. They assume that the topology of cameras is unknown and quite arbitrary, the fields of views covered by these cameras may have no overlap or any amount of independently, and the positions and velocitied of objects along trajectories are computed as features.

3. Proposed Methodology

In order to determine the individual’s status based on “Deviation from Normal Path” factor, a normal path model that is created through training is needed. The normal path model will be applied to the system as based and any path that is taken by the monitored individual that is deviated from the normal path that is already defined in the normal path model will be deemed as suspicious.

Clustering of the similar trajectory is being done by using Dynamic Time Warping (DTW). The DTW algorithm uses a dynamic programming technique to solve this problem. The reason this method is being used is because it is not computational intensive and it can cluster similar trajectory at a very short amount of time. Furthermore, should 2 path with different numbers of points will still be able to be clustered together to form a normal path. The basic requirement for Dynamic Time Warping is the coordinates that is obtained from both the clustered path and also the newly obtained path. We will calculate the distance of each coordinates from both path and each coordinates will be saved in an array and thus 2 arrays are obtained.

We will be using an improved version of Levenshtein Distance to compare both coordinates. Levenshtein Distance is a method that is derived to cluster 2 path with different numbers of points accurately. However, this method is too computationally intensive and thus a new method is required derived from Levenshtein Distance that is both as accurate as Levenshtein Distance but less computational intensive. Using this distance has 2 advantages. The first advantage will be both path does not need to have the same amount of points to be compared and the second advantage is the process of clustering similar path will be made easier. Levenshtein Distance is very computational intensive. If there are 2 paths, where Path S has M points and Path T has N Points, then the number of calculation that needs to be done is M x N times. By using the new and improved method, although both path may not have the same amount of points, a median path can still be obtain by getting the median points of the two nearest point.

After the Human Detection Function is executed, a Tracking Point (TP) is passed into the Path Detection and Clustering Module. This Tracking Point (TP) is the input of this and through this Tracking Point (TP) we can clearly draw a path that is taken by the monitored individual. In order to obtain the Tracking Point (TP), a simple calculation is done. The calculation is shown as follow:-

Coordinate X = Median of Width of Bounding Box

Coordinate Y = Height of Bounding Box * 0.9

Figure 1 shows the breakdown of the process that occurred in the Path Detection and Clustering Module. Once the Tracking Point (TP) is obtained from the Human Detection

Function, a function will be triggered so that a line can be drawn to indicate which path is being taken by the individual that is currently being tracked. This line will be defined as the path that is taken by the individual. This function is being triggered when an individual appears in the video and when the individual goes out from the video, the line that is drawn will be stored into the database and will later on be retrieved to be clustered together with other similar trajectories.

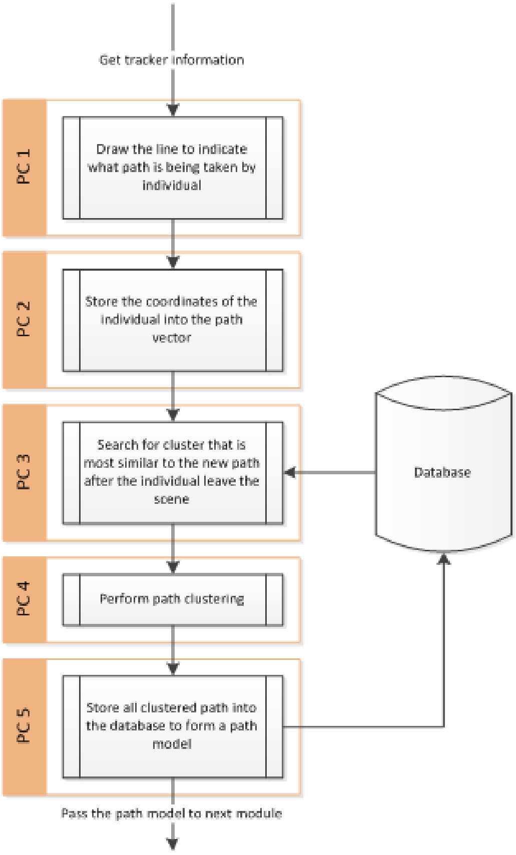
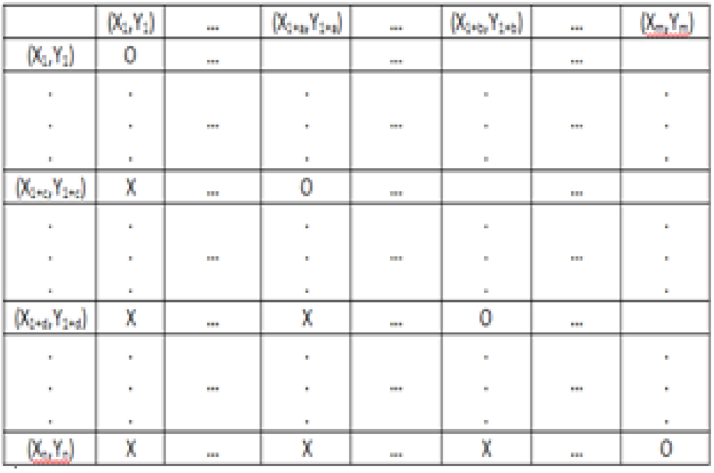

Get tracker information
PC 1
Draw the line to indicate what path is being taken by individual
PC 2
Store the coordinates of the individual into the path vector
PC 3
Search for cluster that is most similar to the new path after the individual leave the scene
Database
PC 4
Perform path clustering
PC 5
Store all clustered path into the database to form a path model
Pass the path model to next module

Figure 1 Path Detection and Clustering Module

Firstly, the first point of the path that is just retrieved will be compared with the entire path in the cluster that is stored in the database in order to only obtain the path that is has similar first point with the retrieved path. By doing so, there is no need to transverse and compare the entire path in the database in order to perform path comparison and clustering. For example, in this experimentation setup that is being carried out in 2 different scene, a total of 6 cluster will be formed after the training phase. However, instead of comparing with all the cluster available, only 2 clusters at most are retrieved and compare with each time. By doing so, the number of processes and computation that is needed to be done will be reduced. Once the similar path has been retrieved from the database, path clustering using Dynamic Time Warping technique will commence.



	(X_1,Y_1)	...	(X_{1+a},Y_{1+a})	...	(X_{1+b},Y_{1+b})	...	(X_m,Y_m)
(X_1,Y_1)	O	...		...		...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_{1+c},Y_{1+c})	X	...	O	...		...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_{1+d},Y_{1+d})	X	...	X	...	O	...	
⋮	⋮	...	⋮	...	⋮	...	⋮
(X_n,Y_n)	X	...	X	...	X	...	O

Figure 2 Calculation of Dynamic Time Warping

There are 2 steps in the Dynamic Time Warping (DTW) technique. The first step is to compare each point in one path with every point in the second path with the second path's starting point begins with wherever the previous best point is found, and by doing so a matrix will be generated. The second step is to work through this matrix, starting at the top-left corner (corresponding to the beginning of both paths), and ending at the bottom-right (the end of both paths): for each cell, the Euclidean Distance is calculated and also ignoring the points from the previous calculation as shown in Figure 2. In Figure 2, the "X" means that that particular point does not need to be calculated and could be ignored while the "O" means that two particular points are the closest to each other.

The method that is used to compute the distance between 2 coordinates is the improved version of Levenshtein Distance. The new path clustering method takes 2 paths, S of length M, and T of length N, and returns the clustered oath between the 2 paths. The 2 paths that is used will be represented with 2 array; s[m] and t[n] where m is the number of points that makes up the path s and n is the number of points that makes up the path t. During the beginning of the process of calculation, the distance for the matrix that is used for clustering calculation will be set to maximum value as shown in Figure 2 by using the formula shown in (1). Once the initial value is set, the calculation can proceed to the next step. Starting from the top-left corner of the matrix, the mth coordinate and the nth coordinate will be compared with each other by calculating the distance between the 2 said coordinates by using the formula shown in (2).

$$d(p,q) = \max(s[1 \ldots m], t[1 \ldots n]) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \ldots\ldots \text{(2)}$$

Once the distance between the 2 coordinates have been computed and the value will be stored and once again the matrix for comparing the 2 coordinates will be consulted. After the smallest value have been found in each row j, the next row will begin it's calculation from the column I where i is the coordinate of the previous smallest value of row j. The process will be repeated until the bottom-right cell is reached. When the 2 points that are closest to each other is determined, they will be clustered together by using the formula shown in (3).

$$Mean\ Path = \left(\frac{Clustered\ \ x + New\ x}{2}, \frac{Clustered\ \ y + New\ y}{2}\right) \ldots\ldots\ldots\ldots\ldots \text{(3)}$$

Clustered = coordinates stored in the clustered path

New = coordinates stored in the inserted path for clustering

The new coordinates that is saved in the Mean Path is the Path Model that is normal. A normal path means that the path that is taken is not suspicious as it is frequently being taken by people.

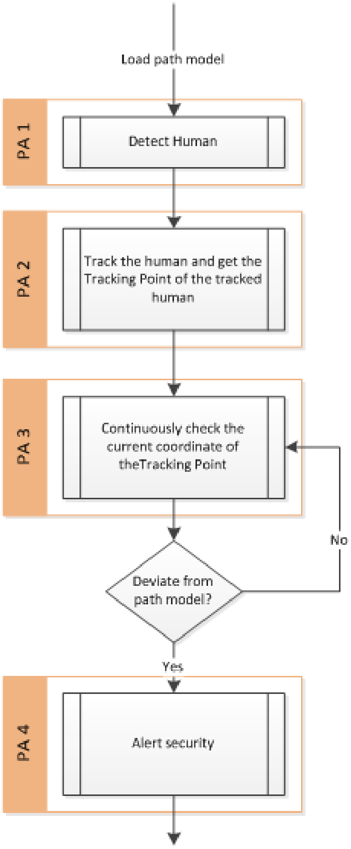

Load path model
PA 1
Detect Human
PA 2
Track the human and get the Tracking Point of the tracked human
PA 3
Continuously check the current coordinate of theTracking Point
Deviate from path model?
No
Yes
PA 4
Alert security

Figure 4 Path Analysis Module

The breakdown of the Path Analysis Module is shown in Figure 4.The Path Analysis Module will determine the status of the currently monitored individual based on deviation from Path Model. In order to determine the individual's status based on "deviation from Path Model" factor, a normal path model that is created through training by Module 2 is needed. The normal path model will be applied to the system as a reference guide and any path that is taken by the monitored individual deviated from the normal will be deemed as suspicious. When the individual is being monitored, the TP is being stored into the database as well.

With the coordinates, a path will be drawn and that path will constantly be compared with the current normal path model's clustered path. If the path that is taken by the monitored individual is still within the Region of Interest (ROI) that is created by thresholding the clustered normal path, then the individual is not suspicious. Otherwise, security personal will be alerted. Figure 5 shows the example of the thresholded clustered path.

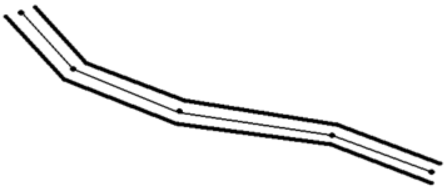


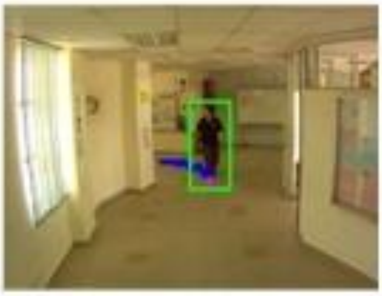
Figure 5 Thresholded Clustered Path

4.Experimental Results

In order to verify the amount of reduced computation during the clustering process, different video sequences were tested. All algorithms are implemented by using a video image size of 320x240 pixels. 2 different scenes are being used to test this algorithm. Figure 6a and Figure 6b are the screen shot of the scene that are being used for testing purposes. Figure 7 is the example of one of the video scene that is used to test the algorithm.

Figure 6a Scene 1

Figure 6b Scene 2



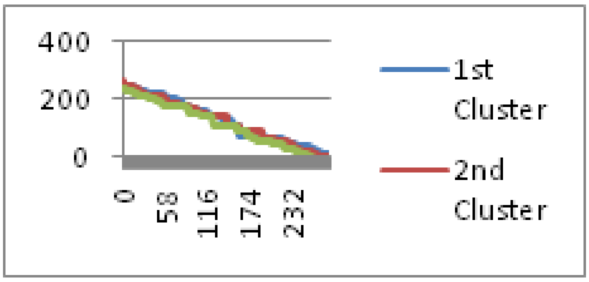
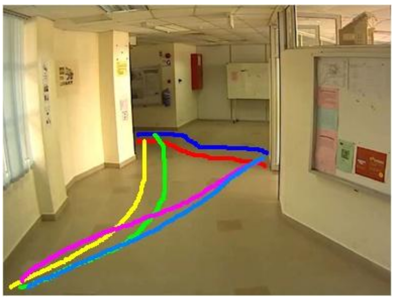


Figure 7 Path Detected

Figure 8 Computation Graph

The Y-axis denotes the number of computation that is required in order to locate the nearest point for clustering while the X-axis denotes the point's position in the main path for clustering. One of the results of the path clustering is shown in Figure 8, the number of computation required is decreasing while searching for the closest point for clustering. By using the cluster index selection method together with the DTW technique, the overall performance of the system has improved.

Trajectory clusters are often represented by routes or paths most commonly taken by motion objects [10]. According to the principle, we chose the scene shown in the Figure 9 and Figure 10.



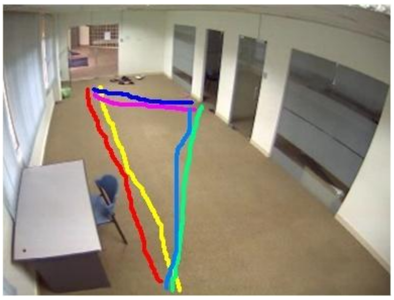


Figure 9 Ground Truth for Scene 1

Figure 10 Ground Truth for Scene 2

All the trajectories that are acquired from 6 different videos that are taken from 2 different scenes are labelled manually with 6 clusters as ground truth in each scene. The average length for the cluster in each scene is 184 points and 189 points respectively. In scene 1, the shortest length of the cluster is 143 points while the longest length of the cluster is 261 points. In scene 2, the shortest length of the cluster is 141 points while the longest length of the cluster is 270 points. Figure 09and Figure 10 illustrate the 6 clusters in both scene 1 and scene 2 where each line denotes a cluster.

Let G be the ground truth set with M clusters, C be the resulting clusters set. Given a cluster C_i that exists in C, the corresponding cluster g_m that exists in G that maximizes $|C_i \cap g_j|$

can be found. Then PCA (Path Clustering Accuracy) is defined to evaluate the accuracy of each cluster by using the formula shown below:-

$$PCA = \frac{1}{N}\sum_{j}^{N}\left(100 - \frac{1}{M}\sum_{i}^{M} P_i\right)$$

N is the total number of clusters formed, M is the total number of points in the cluster and P_i the deviancy between the calculated cluster points in the *i*-th cluster and the expected points in the *i*-th from the ground truth. It is computed as follows:-

$$P_i = \begin{cases} 0 \; if \; \exists C_k \; \in C, C_k \cap g_m = 1 \\ \qquad\quad |C_i \cap g_m| \end{cases}$$

Thus, by using the formula stated above, we can calculate the accuracy of the path clustering by comparing the deviancy rate between the clusters that are formed through the training stage and the clusters that are labelled manually (ground truth). By deducting the deviancy rate with a 100%, the results would be the accuracy of the path clustering. The results from the PCA will be shown in Table where G size is the length of the *i*-th cluster in the ground truth, C size is the length of the *i*-th cluster in the trained cluster, Deviancy Rate is the difference distant between the trained cluster and the expected cluster (ground truth) and Correct Cluster Rate (CCR) is the accuracy of each individual cluster. The average of all the CCR in a scene is the Path Clustering Accuracy (PCA).

Table 1 Path Clustering Accuracy Results

Attribute		G size	C size	Deviancy Rate (%)	CCR (%)	PCA (%)
Scene 1	Cluster 1	228	218	0.88	99.12	99.03
	Cluster 2	261	251	0.79	99.21	
	Cluster 3	168	143	1.01	98.99	
	Cluster 4	156	152	1.14	98.86	
	Cluster 5	157	157	1.01	98.99	
	Cluster 6	143	141	1.01	98.99	
Scene 2	Cluster 1	151	141	0.98	99.02	98.925
	Cluster 2	146	144	1.09	98.91	
	Cluster 3	185	170	1.11	98.89	
	Cluster 4	161	161	1.07	98.93	
	Cluster 5	260	250	1.05	98.95	
	Cluster 6	273	270	1.15	98.85	

From the results shown in Table 1, it is shown that the trained path is quite accurate with an accuracy of 99.03% for scene 1 and 98.925% for scene 2.

5. Conclusions and Future Works

This paper has demonstrated the computation reduction for the path clustering process that is required to generate a Path Model that will determine which path is more frequently taken by passerby that is captured in the scene. Thus, the next step in this research will be to further enhance the accuracy of the system with better path retrieval method and further reduce the number of computation needed.

6. References

[1] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “A Context Based Approach for Detecting Suspicious Behaviors”, IEEE TENCON, 2010.

[2] Kenneth Rose, Member, IEEE, Eitan Gurewitz, and Geoffrey C. Fox “Constrained Clustering as an Optimization Method”, IEEE Journal and Magazine 1993.

[3] Zhang Zhang, Kaiqi Huang, Tieniu Tan, “Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes”, IEEE 2006.

[4] Dimitrios Makris and Tim Ellis, "Path Detection in Video Surveillance", Image and Vision Computing Journal, vol.20/12, pp 895-903, October 2002.

[5] M. Vlachos, G. Kollios, D. Gunopulos, “Discovering Similar Multidimensional Trajectories”, Proc. 18th Intl. Conf. on Data Engineering (ICDE’02), pp.673–685, 2002 .

[6] D.Xie, W.Hu, T.Tan, J.Peng, “A Multi-Object Tracking System for Surveillance Video Analysis”, in Intl Conf. Pattern Recognition (ICPR’04), 2004 .

[7] F.I.Bashir, A.A.Khokhar, D.Schonfeld, “Segmented trajectory based indexing and retrieval of video data”, in Proc. Intl. Conf. on Image Processing (ICIP’03), vol 2, pp.623-626, 2003.

[8] I.Junejo, O.Javed, M. Shah, “Multi Feature Path Modeling for Video Surveillance”, in Proc. 17th Intl. Conf. on Pattern Recognition (ICPR’04), vol 2, pp 716-719, 2004.

[9] F.M.Porikli, T.Haga, “Event Detection by Eigenvector Decomposition Using Object and Frame Features”, in Proc. Conf. on Computer Vision and Pattern Recognition Workshop (CVPRW’04), vol 7, pp 114-121, 2004.

[10] Z.Fu, W.Hu, T.Tan, “Similarity Based Vehicle Trajectory Clustering and Anomaly Detection”, in Proc. Intl. Conf. on Image Processing (ICIP’05), vol 2, pp 602-605, 2005.

[11] Ujjwal Das Gupta, Vinary Menon and Uday Babbar, “Detecting the Number of Clusters During Expectation-Maximization Clustering Using Information Criterion”, IEEE 2010.

[12] R.Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani and M. Ebrahimi, “Vanishing point detection in corridors: using Hough transform and K-means clustering”, IEEE Journal and Magazine 2010.

[13] Neil Johnson and David Hogg, “Learning the Distribution of Object Trajectories for Event Recognition”, British Machine Vision Conference, 2000

[14] Barak Raveh, Angela Enosh, and Dan Halperin, “A Little More, a Lot Better: Improving Path Quality by a Path-Merging Algorithm”, IEEE Journal and Magazine 2011.

[15] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato and Yozo Fujino, “Learning Motion Patterns and Anomaly Detection by Human Trajectory analysis”, IEEE 2007.

[16] Simon Fong and Zhuang Yan, “A security Model For Detecting Suspicious Patterns in Physical Environment”, IEEE, 2007

[17] Simone Calderara, Rita Cucchiara and Andrea Prati, “Detection of Abnormal Behaviors Using A Mixture of Von Mises Distributions”, IEEE 2007.

[18] Simone Calderara, Rita Cucchiara and Andrea Prati, “Using Circular Statistics for Trajectory Shape Analysis”, IEEE 2008.

[19] Daniel Barbara and Maurizio Filippone, “Detecting Suspicious Behaviour in Surveilance Images”, IEEE 2008.

[20] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “Detecting Uncommon Trajectories”, IEEE, 2008

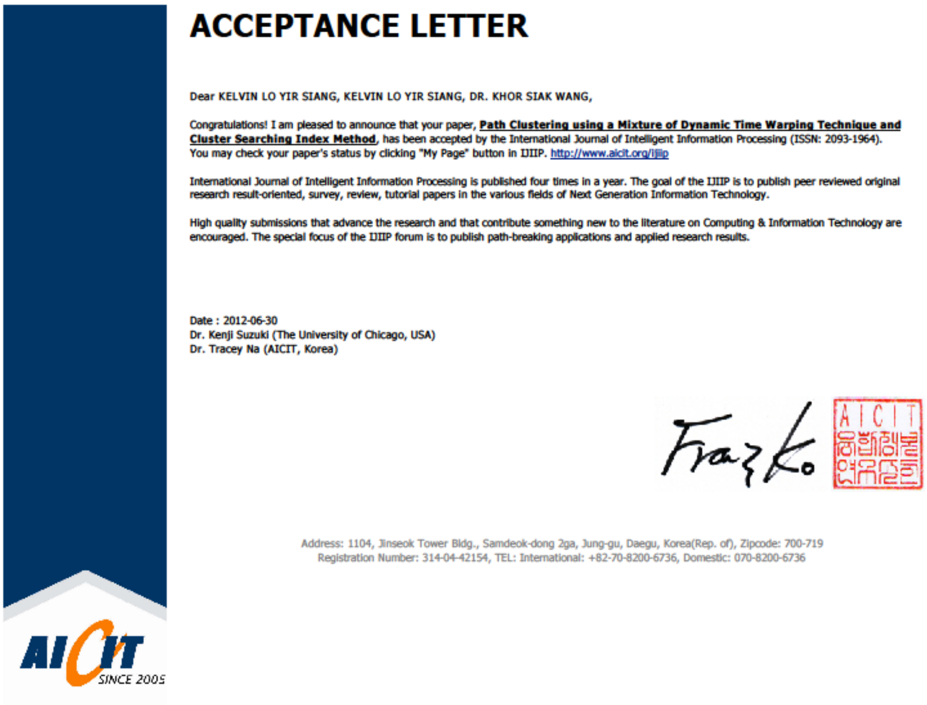
[21] Arnold Wiliem, Vamsi Madasu, Wageeh Boles, and Prasad Yarlagadda, “A Context Based Approach for Detecting Suspicious Behaviors”, IEEE TENCON, 2009.

[22] Yi Feng Zhou and Jim P.Y. Lee, “A Modified Expectation Maximization Algorithm for Maximum Likelihood Direction of arrival Estimation”, IEEE 2010.

[23] Xiaogang Wang, Kinh Tieu, and W. Eric L. Grimson, “Correspondence-Free Activity Analysis and Scene Modeling in Multiple Camera Views”, IEEE Journal and Magazine 2010.

Appendix F

PUBLICATION – LETTER OF ACCEPTANCE FOR MANUSCRIPT IN Appendix E



ACCEPTANCE LETTER

Dear KELVIN LO YIR SIANG, KELVIN LO YIR SIANG, DR. KHOR SIAK WANG,

Congratulations! I am pleased to announce that your paper, **Path Clustering using a Mixture of Dynamic Time Warping Technique and Cluster Searching Index Method**, has been accepted by the International Journal of Intelligent Information Processing (ISSN: 2093-1964). You may check your paper's status by clicking "My Page" button in IJIIP. http://www.aicit.org/ijiip

International Journal of Intelligent Information Processing is published four times in a year. The goal of the IJIIP is to publish peer reviewed original research result-oriented, survey, review, tutorial papers in the various fields of Next Generation Information Technology.

High quality submissions that advance the research and that contribute something new to the literature on Computing & Information Technology are encouraged. The special focus of the IJIIP forum is to publish path-breaking applications and applied research results.

Date : 2012-06-30
Dr. Kenji Suzuki (The University of Chicago, USA)
Dr. Tracey Na (AICIT, Korea)

AICIT

Address: 1104, Jinseok Tower Bldg., Samdeok-dong 2ga, Jung-gu, Daegu, Korea(Rep. of), Zipcode: 700-719
Registration Number: 314-04-42154, TEL: International: +82-70-8200-6736, Domestic: 070-8200-6736

AICIT SINCE 2005