

國立台灣大學生農學院生物產業機電工程學系

碩士論文

Department of Bio-Industrial Mechatronics Engineering

College of Bioresources and Agriculture

National Taiwan University

Master Thesis

合作式影像系統在農業環境作業模式分析之應用

Applications of Cooperative Imaging System on Working

Pattern Analyses in Agricultural Environment

黃揚文

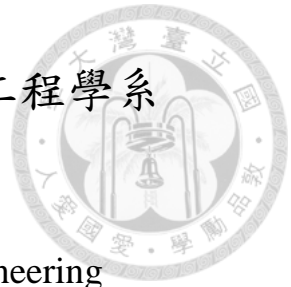
Yang-Wen Huang

指導教授：林達德 博士

Advisor: Ta-Te Lin, Ph.D.

中華民國105年8月

August 2016



國立臺灣大學碩士學位論文
口試委員會審定書

合作式影像系統在農業環境作業模式分析之應用
Applications of Cooperative Imaging System on
Working Pattern Analyses in Agricultural
Environment

本論文係黃揚文君（學號 r03631012）在國立臺灣大學生物
產業機電工程學系、所完成之碩士學位論文，於民國 105 年 07
月 07 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

林達德

（簽名）

（指導教授）

史祥

陳世芳

系主任、所長

鄭宗記

（簽名）

誌謝



終於到了畢業時分，掐指一算，在林達德教授的實驗室也待了五年，回想起當初大一，很幸運地遇到程式語言設計的助教笑點學長，啟發了我對於程式相關領域的興趣，於是我來到了這裡，與實驗室的夥伴一同參加比賽、一同吃喝玩樂。

首先我想感謝林達德教授的教誨，他點足了我研究上的一些盲點，也同時給予我研究上分析表現的方法以及目標，讓我能夠將我的研究更清楚的呈現；感謝艾群教授提醒以及建議我的研究的真正應用之處，以及同樣能夠應用的相關領域，讓我能把程式設計的更接近使用者的預期；也謝謝陳世芳助理教授，告訴我如何撰寫論文，還有一些研究方法上的建議。

最要感謝的是，陪著我做完所有實驗的助理詹川億，沒有你的幫忙，我實驗的架設的時間及困難一定會大大上升，也謝謝你願意陪我北中南台灣到處奔波。也謝謝實驗室兩位同學：徐嘉君、蔡靜偉兩年來相挺，因為年齡相仿又為同學，能夠在我盲目的時候提出許多較為相近不失真的建議，真的助益良多。感謝大我一屆的三位學長家瑋、阿剛、榕修多年的陪伴，教我許多實驗室的事情以及課業上的幫助，也謝謝志軒學長、金毛學長陪伴我討論一些課業上的難題。

中文摘要



合作式影像系統是一個用於解決長時間觀察的方法。據此，我們建立了一套合作式影像系統用來幫助農業環境中的管理者或是擁有人能夠更了解他們的設施運作狀況、以及一些需要長時間觀察才能獲得的資訊。

本研究繼承了前代系統的架構並進行改良，舊系統採用環場影像以及 Pan-Tilt-Zoom (PTZ) 攝影機的個別優勢，將其組合在一起同時取得廣角的全域影像以及局部的高解析度影像；由於硬體效能上的限制以及環場影像攝影機的畫面品質較為不足，因此新系統使用了廣角攝影機以及 PTZ 攝影機，其中廣角攝影機之角度高達 135 度，能夠捕捉該地大部分所發生的事件影像，再利用能夠自由進行光學變焦的 PTZ 取得欲觀察之局部高解析影像。

在軟體方面架構上我們從主從式影像系統改成合作式影像系統，讓攝影機由單純的主從關係轉變成對等並能夠進行情報的交換，有助於個別攝影機在下次能夠更容易地尋找到所要觀察之目標。為偵測所有活動區域的目標，尋找前景的部分使用了多重解析度高斯混合模型背景相減法 (Multi-resolution Gaussian Mixture Model) 和靜態物體追蹤法 (Static Object Detection)，並使用動態歐式距離 (Dynamic Euclidean Distance) 對抓取目標的演算法閾值進行自動化參數調整，為了使每個攝影機都能夠獨立運作並進行目標的追蹤，系統改為多重並序的架構進行運作，並且利用 PTZ 影像進行像差 (frame differential) 與質心預測 (centroid estimation) 演算法讓 PTZ 也能夠自動進行物件追蹤以增加抓到目標細節影像的準確率。這些來自攝影機的資料將會被送到系統後端進行分析產生工作模式分析，辨識物體的類別是透過由加柏濾波器 (Gabor Filter) 和詞袋模型 (Bag of Words) 組成的混合特徵擷取演算法進行追蹤物件的擷取，再透過支持向量機演算法 (Support Vector Machine, SVM) 分析結果，為了讓系統能夠在不同環境下使用，我們建立了一套自定義規則系統，讓使用者能夠使用軌跡資訊以及支持向量機演算

法分類的類別對該地進行組態的客製化以符合需求，並且提供使用者一個良好的使用者介面去觀察分析的結果。

本研究進行了五次實驗以驗證系統的有效性，第一次實驗在台大園藝系溫室驗證並改進了物體追蹤演算法；第二次實驗在紫城農場進行農業環境簡單條件下的測試，透過該實驗設計並改良自定義追蹤系統；第三次實驗在第三班蔬菜集貨包裝廠測試農業環境下集貨區的工作模式，確保系統能夠有環性的自適性；第四次實驗在台大知武館，強化了 PTZ 自我追蹤演算法的架構；第五次實驗在農機館門口，再次對系統做了驗證並且改進了系統的資料架構以及系統架構的多重並序。我們開發的合作式影像系統能夠確保在事件發生時可以取得最大的資訊量以便我們進行精準的觀察與分析。

關鍵字：合作式影像系統、模式分析、物件追蹤、自定義規則系統、農業環境

Abstract



The surveillance systems solve the difficulties of long-term observation. We are designing a cooperative surveillance system to make managers and owners get to know their places of the agricultural environments better by providing the working patterns and the other additional long-time observed information.

This research continues the predecessor's system structure and improves it. The old system use a combination of a panorama camera set and a Pan-Tilt-Zoom (PTZ) camera. The system has the advantage of monitoring the object's surroundings and the object itself in high resolution at the same time. Due to the limitation of the hardware performance and the low image quality from the camera set, the new system uses an ultra-wide field of view (FOV) camera and a Pan-Tilt-Zoom (PTZ) camera. Ultra-wide FOV images from the static camera up to 135 degree provide most of any possible happenings, and the images from PTZ camera fill the information of the low-resolution images of the ultra-wide FOV camera.

In the software, we changed the system from Master-slave system to Cooperative Surveillance system. The relationship between the cameras has changed from inequality to equal. The Network Control Center in the system is able to let the cameras to communicate with each other instead of doing their works on their own. In order to detect the objects from the cameras, we use Multi-resolution Gaussian Mixture Model, Static



Object Detection and Dynamic Euclidean Distance to adapt the video sequence in different environment. Some changes are made to make the system more “cooperative”.

First, the system structure is changed to parallel thread processing. We also use frame differential and centroid estimation to process the PTZ images to make the PTZ able to do self-tracking tasks to increase the accuracy of capturing any tracked object. All the information from the cameras will be sent to the Network Control Center to analyze the working patterns of a place. We use mixture feature extracting method consist of Gabor filter and Bag of Words to process the images of the detected objects. The processed featured will be trained and predicted using SVM. To make the system able to fit different environment use, a Custom Define Rules system is provided to let users to create their own working patterns by existing features and the trajectory information. An Object View Manager is also provided for users to look up the individual detail of any detected object.

We designed five experiments to validate our system. The first experiment is done in the NTU farm; we verified our object tracking methods with the new camera. The second experiment is done in the Zhi Chen farm. We enhanced and redesigned the Custom Define Rules System. The third experiment in the 3rd vegetable packaging factory verified the previous changes is adaptable in different environment. With the fourth experiment outside of the Tomatake Hall, we improved the PTZ self-tracking algorithm. At last, the fifth experiment in the plaza in front of the Dept. of the BIME, we again verified the

above changes and made a final improvement of our system and data structure. The cooperative system ensures the maximum information during any event that provides us to judge more precisely.

Keywords: Cooperative Surveillance System, Pattern Analyses, Object Tracking, Custom

Define Rules System, Agricultural Environment

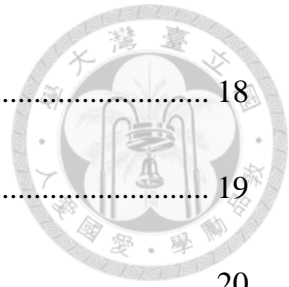


Content



中文摘要	i
Abstract.....	iii
Content.....	vi
Content of Figures.....	x
Content of Tables.....	xv
Chapter 1 Introduction	1
1.1 General Background Information.....	1
1.2 Research Objectives	4
Chapter 2 Literature Review	6
2.1 Surveillance System	6
2.2 Multi-camera Setup	6
2.3 Cooperative Surveillance System.....	8
2.4 Agricultural Environment Related Works	10
2.5 Object Detection.....	11
2.5.1 Gaussian Mixture Model Background Subtraction	12
2.5.2 Fast Self-Tuning Background Subtraction Algorithm.....	13
2.5.3 Building PTZ Background Subtraction Model.....	15
2.5.4 Histogram of Oriented Gradients for Human Detection	17

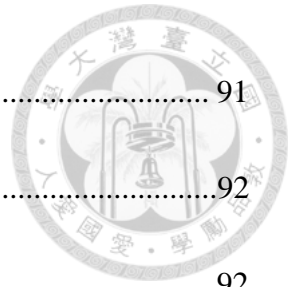
2.6	Object Tracking	18
2.6.1	Bayesian Filtering.....	19
2.6.2	Mean-Shift Object Tracking	20
2.6.3	Kalman Filtering.....	22
2.6.4	Particle Filtering	23
2.7	Behavior Recognition.....	24
2.7.1	Gabor Filtering	25
2.7.2	Least Squares Support Vector Machine.....	26
2.7.3	Neural Network	28
2.7.4	K-Nearest Neighbor.....	30
2.7.5	Naive Bayes.....	30
Chapter 3	Materials and Methods	32
3.1	System Architecture.....	32
3.1.1	Hardware Design	32
3.1.2	Software Design	35
3.1.3	Data Structure	36
3.2	Cooperative system	38
3.3	Object Detection and Tracking.....	39
3.3.1	Gaussian Filter and Interactive Multiple Model (IMM) Filter	39





3.3.2	Geometry Transform between Cameras	42
3.3.3	PTZ Image Mapping to PTZ Sphere coordinate	44
3.3.4	PTZ Self-Tracking Algorithm.....	45
3.3.5	Human Detection and Human Face Detection	49
3.3.6	Simple Static Object Detection (SOD).....	50
3.3.7	Mixture of Static Object Detection and Object Tracking	56
3.3.8	Assignments of the Static Camera and PTZ.....	59
3.3.9	Software Acceleration.....	59
3.4	Data analysis.....	61
3.4.1	Choosing Working Patterns	61
3.4.2	Finding Usable Features	65
3.4.3	Mixture Behavior Features (Gabor and Bag of Words).....	72
3.4.4	Feature Classifications.....	73
3.4.5	Custom Define Rules System.....	74
3.4.6	Working Pattern Analysis and Daily and Weekly Summary	83
3.4.7	Object View Manager.....	83
3.5	Experiment	85
3.5.1	Experiment Sites.....	85
3.5.2	Object Tracking	90

3.5.3	Static Object Detection.....	91
Chapter 4	Results and Discussion.....	92
4.1	System Performance.....	92
4.1.1	Simple Object Tracking.....	92
4.1.2	PTZ Image Mapping to PTZ Sphere Coordinate	94
4.1.3	PTZ Self-tracking Algorithm.....	97
4.1.4	Acquired Images from PTZ.....	99
4.1.5	Human Face Detection by PTZ	105
4.1.6	Static Object Detection.....	106
4.1.7	Mixture of the Static Object Detection and Object Tracking	109
4.2	Data Analysis.....	110
4.2.1	Choosing Machine Learning Method.....	110
4.2.2	Overall Accuracies and Misdetections	121
4.2.3	Working Pattern Analysis	124
4.3	Software Performance	146
Chapter 5	Conclusions and Suggestions	147
5.1	Conclusions	147
5.2	Suggestions.....	150
References...	151



Content of Figures



Figure 2-1 Master-slave system.....	7
Figure 2-2 Pseudo code of building a PTZ fixed-view background subtraction model. 15	
Figure 2-3 Representation of PTZ background remapping	16
Figure 2-4 One node in a Neutral Network	29
Figure 2-5 Different layers of Neural Network	29
Figure 3-1 Picture of the Chosen Hardware	32
Figure 3-2 Software GUI design of the Cooperative Surveillance System.....	36
Figure 3-3 Data Structure Diagram of Our Cooperative Surveillance System	36
Figure 3-4 Diagram of Network Control Center	38
Figure 3-5 Illustration of Dynamic Euclidean Threshold.....	40
Figure 3-6 Pseudocode of Data Association and Correction.....	41
Figure 3-7 IMM model for Network Control Center (NCC).....	42
Figure 3-8 Illustration of Geometry Transform.....	43
Figure 3-9 Pseudo code of finding the centroid of the tracked object.....	46
Figure 3-10 Centroid estimation process of PTZ self-tracking.	47
Figure 3-11 The PTZ tracking strategy of the previous system	48
Figure 3-12 New PTZ tracking strategy of the system.....	49
Figure 3-13 Diagram of Static Object Detection.....	51

Figure 3-14 Noise Comparison between normal and morphology-fix.....	52
Figure 3-15 The process of the SOD algorithm	53
Figure 3-16 The image during the initialization stage of each algorithm	55
Figure 3-17 False alarm effect on fast self-tuning algorithm.	56
Figure 3-18 Object tracking process without SOD	57
Figure 3-19 Object tracking process with SOD	58
Figure 3-20 The working patterns found in Zhi Chen farm	62
Figure 3-21 The working patterns in the 3 rd vegetable packaging factory in Taoyuan County	63
Figure 3-22 The working patterns in the outside of the Tomatake Hall	64
Figure 3-23 The working patterns in the Plaza of the Dept. of BIME	65
Figure 3-24 A severely jittered path	69
Figure 3-25 The pre-defined map points in Zhi Chen farm	70
Figure 3-26 The setting panel of the pre-defined map points.....	71
Figure 3-27 An example of a tracked sub-image, labled “WithBox”	71
Figure 3-28 Acquired sub-mages with different resolutions	72
Figure 3-29 Different image size effect on Bag of Words.....	72
Figure 3-30 Fast prototyping and training using Orange	73
Figure 3-31 The setting panel of the custom define rules	74



Figure 3-32 GUI of the Object View Manager 84

Figure 3-33 The panorama view of the NTU greenhouse 86

Figure 3-34 The view of the static camera in the system 86

Figure 3-35 Two cases of the Tomatake Hall 87

Figure 3-36 The scene and the setup of the PTZ self-tracking algorithm 88

Figure 3-37 The experimental site of Zhi Chen farm 89

Figure 3-38 The experimental site of the 3rd vegetable packaging factory 90

Figure 4-1 Simple object tracking demonstration 92

Figure 4-2 Errors in Different Zooming Rate..... 94

Figure 4-3 Errors caused by the tangent effect..... 95

Figure 4-4 Threshold representation of the tilt angle for each error rate..... 96

Figure 4-5 The error rate of angle threshold for distance about 350 pixels of different tilt angle 97

Figure 4-6 Example of PTZ Self-tracking algorithm 98

Figure 4-7 Image captured from PTZ..... 99

Figure 4-8 The tracked object in static camera and its PTZ image in Zhi Chen farm . 100

Figure 4-9 Fail case when zoom in object is not at the center of the PTZ image 101

Figure 4-10 PTZ self-centered the detected crowded objects and zoom in to the center of the image 102



Figure 4-11 Tracing single object with PTZ.....	103
Figure 4-12 Tracing multiple objects with PTZ	104
Figure 4-13 The acquired faces from the PTZ image.....	105
Figure 4-14 The results of the static object detection.....	107
Figure 4-15 False alarm demonstration of the SOD algorithm	109
Figure 4-16 Features chosen to classify between “with box” and ”no box”	110
Figure 4-17 Water reflection (left) and shadow hazard (right).....	123
Figure 4-18 The daily result from Zhi Chen farm of the first day data.....	125
Figure 4-19 Weekly summary of Zhi Chen farm.....	125
Figure 4-20 Daily summary from 2016-03-10 to 2016-03-16 of Zhi Chen farm.....	128
Figure 4-21 Path usage (left) and the most stay positions (right) of the Zhi Chen farm	129
Figure 4-22 Pre-defined map points of Zhi Chen farm	130
Figure 4-23 The daily result from 3 rd vegetable packaging factory of the first day data	131
Figure 4-24 Weekly summary of 3 rd vegetable packaging factory	132
Figure 4-25 Daily summary from 2016-04-27 to 2016-05-03 of 3 rd vegetable packaging factory.....	135
Figure 4-26 Most stay positions of 3 rd vegetable packaging factory.....	136



Figure 4-27 Pre-defined map points of 3rd vegetable packaging factory 137

Figure 4-28 Daily summary outside of the Tomatake Hall 138

Figure 4-29 Weekly summary outside of the Tomatake Hall 139

Figure 4-30 Daily summary from 2016-06-23 to 2016-06-29 of outside the Tomatake
Hall 141

Figure 4-31 Most stay position of the road outside of the Tomatake Hall 142

Figure 4-32 Pre-defined map points outside of the Tomatake Hall..... 143

Figure 4-33 Daily summary of the plaza in front of the Dept. of BIME..... 144

Figure 4-34 Most Stay Position of the plaza in front of the Dept. of BIME 145

Figure 4-35 Pre-defined map points of the plaza in front of the Dept. of BIME 145

Content of Tables



Table 3-1 Specification of SNC EB-630	33
Table 3-2 Specification of Theia SY125M.....	34
Table 3-3 Specification of ACTi® CAM-6630 High Speed Dome Camera	35
Table 3-4 Data Structure of the Cooperative Surveillance System	37
Table 3-5 Working patterns represented by the combination of the features in Zhi Chen farm.....	66
Table 3-6 Working patterns represented by the combination of the features in 3 rd packaging factory	67
Table 3-7 Working patterns represented by the combination of the features outside of the Tomatake Hall.....	68
Table 3-8 Working patterns represented by the combination of the features in the Plaza in front of the Dept. of BIME.....	68
Table 3-9 Pre-defined map points of Zhi Chen farm.....	77
Table 3-10 Custom rules for Zhi Chen farm.....	77
Table 3-11 Pre-defined map points of the Packaging Factory.....	78
Table 3-12 Custom rules for Packaging Factory	79
Table 3-13 Pre-defined map points of the Tomatake Hall.....	80
Table 3-14 Custom rules for Tomatake Hall.....	81

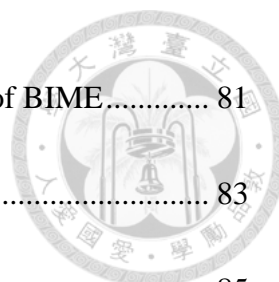
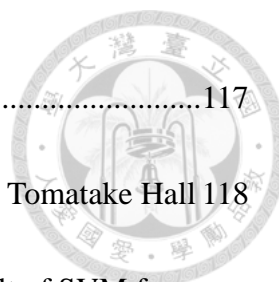


Table 3-15 Pre-defined map points of the plaza in front of the Dept. of BIME.....	81
Table 3-16 Custom rules for the Dept. of BIME.....	83
Table 3-17 Experiment sites and the purposes	85
Table 4-1 The disconnect rate comparison between fixed and dynamic Euclidean threshold	93
Table 4-2 The results of static object detection	106
Table 4-3 The performance of the Mixture of SOD and Object Tracking.....	110
Table 4-4 Gabor filter. 2-Feature training results of Zhi Chen farm.	111
Table 4-5 Gabor filter. Confusion matrix of the feature training results from Zhi Chen farm.....	112
Table 4-6 Bag of Words. 2-Feature training results of Zhi Chen farm.....	113
Table 4-7 Bag of Words. Confusion matrix of the feature training results.....	113
Table 4-8 Gabor filter. 2-Feature training results from 3 rd vegetable packaging factory	114
Table 4-9 Gabor filter. Confusion matrix of the feature training results from 3 rd vegetable packaging factory.	115
Table 4-10 Bag of Words. 6-Feature training results from 3 rd vegetable packaging factory.	116
Table 4-11 Bag of Words. Confusion matrix of the feature training results of SVM from	



3 rd vegetable packaging factory.....	117
Table 4-12 Gabor filter. 2-Feature training results from outside of the Tomatake Hall	118
Table 4-13 Gabor filter. Confusion matrix of the feature training result of SVM from outside of the Tomatake Hall.....	118
Table 4-14 Gabor filter. 2-Feature training results from plaza in front of the Dept. of BIME	119
Table 4-15 Gabor filter. Confusion matrix of the feature training result of SVM from plaza in front of the Dept. of BIME	119
Table 4-16 The overview of all the accuracies	120
Table 4-17 Overall accuracy of the vegetable packaging factory using raw trajectories sub-images	122
Table 4-18 Misdetections cause by noise and disconnected trajectories.....	123
Table 4-19 The performance comparison between the old system and the new system	146

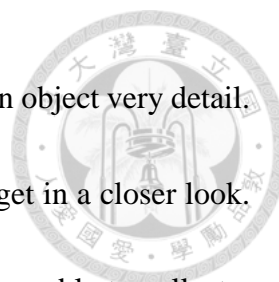


Chapter 1 Introduction

1.1 General Background Information

The widespread of surveillance systems helps investigating the crime scenes and car accidents. Monitoring systems solve the difficulties of long-term observation. The data provide the changes, the people, and other valuable information. However, finding useful data is not easy. Checking these videos manually takes up too much time. To simplify the data, many people put effort in image processing related work. These image-processing techniques can automatically analyze and send the desired data to the users only. For example, an application of unattended or stolen objects detection could be useful in public areas (San Miguel and Martinez, 2008). For traffic control, road monitoring identifies anomalous behavior (Feizi et al., 2013). Another application of health care uses IR cameras and IR pattern projectors for posture detection (Utsumi et al., 2006).

There are two major obstacles of a RGB camera. The most important issue is the size of the viewable area. Common cameras have a view angle of 110 degrees. Instead of deploying multiple cameras in several locations, we use a network camera with exchangeable lens. The view angle of lens is 135 degrees, and it has physical tuned curvature that makes the view without distortion. The second problem of a single RGB camera is the insufficient information of small objects on the screen. A group of pixels describes every objects on the screen. If the size of the object is too small or the resolution



of the camera is too low, we will not have enough pixels to describe an object very detail.

We add a Pan-Tilt-Zoom camera (PTZ) to track and magnify our target in a closer look.

The combination of ultra-wide lens camera and PTZ makes the system able to collect a wide-view image without losing any target details. The combination system is called master-slave surveillance system.

Although a master-slave system has the advantage of costs and information-gathering capabilities, it also has drawbacks. Master-slave systems are more complex than a single camera system, which needs a better PC to run the system. Its image might be distorted because the lens edge and the CCD effect according to different cameras and lens. We have to put more effort in building the relation between the static camera and the PTZ camera. By using the method proposed by 余 (2012), where homography calibration method (Chen et al., 2008) was used in a webcam based panorama system (余, 2012), we describe the related position of the camera models using polynomial. We correct the system using the real world information the make the system more accurate. Another drawback of the master-slave system is that the master camera (static camera) always controls the slave camera (PTZ camera) instead of communicating to each other. A cooperative smart-camera (Micheloni et al., 2010) was introduced. This system also has a static camera and one or more PTZ cameras. The only difference between them is that instead of letting the static camera controls the PTZ camera, the cooperative camera

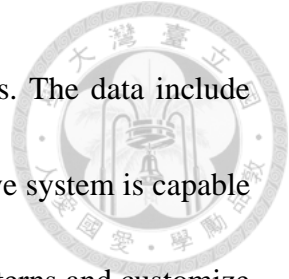


system builds a network control system. This system controls all the cameras, and the cameras control the system. The algorithms inside the network system auto select the best result according to every input of the cameras. The cooperative system is more robust and more intelligent. By letting the cameras “talk” to each other surely makes the tracking easier. Any camera that detects the object can notice the network system to send the location to tell the others.

When finding any object in the panorama, we use geometric transformations to setup the accurate position on the PTZ camera by creating a control center in the system. However, the object we found might move around and leave the position when the time static camera system sends the coordinate to the PTZ camera. Thus, we will have to predict the objects movement and send the predicted position to the PTZ camera, and we will make the PTZ camera has the ability to track the objects on its own and send the results back to the control center. The PTZ itself also has the ability to track object on its own. The PTZ and the static camera will record the behavior of the object, and save the results to the database. At last, using machine learning and geometry methods to analysis and summarize the working pattern of the place. The information helps managers and owners get to know their facilities better.


We design a cooperative surveillance system to monitor the behaviors in each task and analyze the working patterns in agricultural environment. By using the image

processing techniques, we are able to obtain more data for analysis. The data include trajectory, time, and visual sight of any object. Best of all, cooperative system is capable of long-term observation. This system helps us to summarize the patterns and customize the monitoring mode of a place.



1.2 Research Objectives

We are trying to design a cooperative surveillance system that can replace the long-term observation with human eyes. The system is able to automatically analysis the working patterns in intelligent ways of a place comparing to other surveillance systems. It is an all-in-one system from observations to the analysis results. The system is able to collect data in a more efficiently, and more robust way with the minimum side effects. The combination of the ultra-wide FOV camera system and PTZ camera has the advantage of monitoring the object's surroundings and the object itself in high resolution at the same time. Ultra-wide FOV images from the static camera up to 135 degree provide most of any possible happenings, and the images from PTZ camera fill the information of the low-resolution image of the ultra-wide FOV images. The control center in the system is able to let the cameras to communicate with each other instead of doing their work on their own. The cooperative system ensures the maximum information during any event that provides us to judge more precisely.



This research continues the predecessors that improves the hardware design, the tracking methods, the software design, and the improvement of the success rate. We are targeting to develop a system to analyze the working patterns in agricultural environments and provide feedback to the database for future detection and analysis in public areas.

The major purposes include:

1. Develop a system that is able to summarize and give reports periodically of working patterns of a place.
2. Provide adjustable feature classification and geometry definitions to fit the needs for any user.
3. Build a control center in the system to facilitate the communication between cameras in bi-directions.
4. Redesign the software and the hardware to make the system faster and smaller.

Chapter 2 Literature Review



2.1 Surveillance System

Surveillance systems help people ensure the safety of public areas. Some of them are widely used in highways to track and analysis the traffic, and others are used in anti-theft monitoring. With the development of the image processing, people start merging intelligent into the system to auto detect and analysis for us. For example, the application of finding unexpected human behavior in image sequences uses compound features such as accumulated map, crowd pace, and crowd density (Zweng and Kampel, 2010). The system not only detects the abnormal but also provides the habitation of a place.

2.2 Multi-camera Setup

A single camera is not easy to gather all the information we need. Hardware fusion helps us to get more data. For example, Utsumi et al. (2006) use IR cameras to build 3-D appearances of a human and use CCD camera to observe the human behavior. There are also many related works using multi-camera, but most of them are all static cameras or dynamic cameras. Static cameras are fixed and the field of view is limited to the hardware and the placed location. Dynamic cameras (Stillman et al., 1998) have too many variable factor that make them hard to monitor a large area. The combination of the static and dynamic camera system makes us not only able to monitor a large area but also zoom in for a high-resolution image of a detected object.

The Distant Human Identification (DHID) system is a master-slave, real-time surveillance system designed by Zhou et al. (2003). When the master camera detects a moving person, the slave camera is commanded to turn to that direction in zoom-in images. The master is a static camera and the slave is a pan-tilt-zoom camera (PTZ camera). The images of the two cameras is corresponded according the coordinate geometry of the static and PTZ cameras.



Figure 2-1 Master-slave system (Zhou et al., 2003).

Nevertheless, this algorithm requires a priori knowledge of the omnidirectional camera's projection model to solve the nonlinear correspondences between cameras. Chen et al. (2008) propose two methods to overcome the limitation: geometry and homography calibration. A set of polynomials is used to directly relate the master camera coordinate (x_i, y_i) and the PTZ camera coordinate $(\theta_{i,P}, \theta_{i,T}, f_i)$. The correlation of the two coordinates is estimated by the Pearson product-moment coefficient. The methods

increase the accuracy of the geometry transformation, and decrease the complexity of the initial setup.




It is always better to have the maximum field of view (FOV) in the master camera.

Zhou et al. (2003) use a wide-angle lens, while Chen et al. (2008) use a fisheye for a larger FOV. However, fisheye lens distort image, and make the object harder to detect and track. 余 (2012) uses a 360 degree panorama system built with 8 webcams in circle instead. The images of the webcams are stitched using Speeded Up Robust Features (SURF)(Bay et al., 2008) and RANdom Sample Consensus (RANSAC)(Fischler and Bolles, 1981) algorithms. SURF matches the corresponding points between images and RANSAC removes the wrong matching points.

2.3 Cooperative Surveillance System

With the advantage of the PTZ camera, we successfully solved the problems such as increasing the solution of the observed objects. There are three different architectures for network exploiting PTZ cameras: master-slave, autonomous PTZ, and cooperative smart cameras (Michelsoni et al., 2010). The static camera usually controls the PTZ in the master-slave system. The PTZ in the autonomous PTZ controls itself by analyzing what the PTZ sees. Cooperative smart cameras are different from the two above. All the PTZ cameras are controlled by a network system, and every cameras including the PTZ cameras are the inputs of the network system. The event is analyzed by processing the



streams coming from the inputs. Chong et al. (2012) proposed a distributed optimization strategy, the camera gain by reducing the error covariance of the tracked object or through high-res feature acquisition. The cameras negotiate by the track algorithm and are feedback with the user-defined criteria. This makes the system more intelligent, for example, Ghidoni et al. (2010) use a dual camera sensor to track the moving objects and face detection in a cooperative way. Another application of tracking a fast moving object (Su Jeong et al., 2014) is to use three cameras to detect, calculate the position, and capture the moment of the detection time of an object. The information then is sent to the server to determine the speed of the average velocity and its trajectory.

Cooperative system not only contains only a static camera and a PTZ camera. Yan and Gu (2014) use a gun-dome system to automatically realize acquisition, refinement and fast retrieval of the target information. The gun camera and the dome camera determine the initial view of the dome camera itself. Jingqing and Xi (2015) use not only 2D wireless camera sensor networks (WCSNs) information but also add 3D WCSNs. They developed a multi-hop cooperative multi-input-multi-output and orthogonal frequency-division multiplexing (MIMO-OFDM) algorithm to fulfill their needs on searching objects.

2.4 Agricultural Environment Related Works



Most of the information of agricultural environment come from sensors, especially wireless sensor networks installed in the greenhouse. In the past, the thing that people care is to monitor the condition of the environment to increase the output value of the plant such as CO₂ measurement (Tamaki et al., 2015). Some use to trace down the local climate parameters (Ahonen et al., 2008), and some build a system to provide real time monitoring of the important factors in plant growth such as the carbon dioxide, temperature, humidity level in greenhouse (Saad et al., 2014). Despite most of the sensor-related works focus on the plant in the agricultural facilities, there are works that try to build a sensor network to locate and observe the working pattern of farmers. Hashimoto et al. (2016) use radio waves technics to locate the farmer's position by his own cellphone and the beacons installed in the greenhouse.

There are also some researches use camera to reach the goal. Most of the works use one single camera installed on a robot and is use for map building, plant cares and harvesting. Yang et al. (2007) use a stereo vision to recognize tomato for automatic harvesting. Li et al. (2009) use a binocular stereo vision technique for 3D positioning to spray the plants in the greenhouse. Xia et al. (2015) use Kinect to build 3D plant leaf maps. There are also interesting researches such as using linear offset interference to detect the temperature from cameras (Wenjing and Xueqiang, 2014).

2.5 Object Detection



There are hundreds of ways to detect an object using image processing and machine learning technics. Machine learning methods train pre-processed features using classifiers as prior knowledge. Then the trained databases are used as templates to fit the unknown object into the best category, and the answers are returned. Image processing methods find a common answer using algorithms and detect the object with the algorithm. The most common object detection for an image sequence are frame-related algorithms, such as frame subtraction, temporal differencing, and background subtraction model. Frame subtraction uses the pixel difference between frames as an error image. The image is thresholded after the average error is calculated. The algorithm is simple but is extremely sensitive to lights, and apertures. Temporal Differencing is similar to frame subtraction but uses the pixel difference of the image sequence instead of the single images. The object will be treated as a moving object if the subtracted value is greater than the threshold. This algorithm is very sensitive to environment changes, and may produces large amount of fragments that will affect the detection results. Background subtraction model builds the background information of the image sequence and the objects not belong to the background will be treated as moving objects. Because background subtraction model algorithms are good at resisting noise, and less sensitive to light compare to the previous two methods, these kind of algorithms are widely used in



detecting moving objects. Background subtraction model can be further categorized into static and non-static camera models; the following algorithms are static models.

2.5.1 Gaussian Mixture Model Background Subtraction

This algorithm builds N Gaussian mixture models. The distribution of the variable is described by the average and standard deviation, and the distribution shape is described by the covariance matrix. The Gaussian mixture model is shown in equation 2.

$$P(X_t) = \sum_{i=1}^K W_{i,t} \eta(X_t | \mu_{i,t}, \sigma_{i,t}), \quad (2.1)$$

where $W_{i,t}$ is the i^{th} Gaussian distribution weight, and $\eta(X_t | \mu_{i,t}, \sigma_{i,t})$ is the i^{th} Gaussian distribution. $\mu_{i,t}$ and $\sigma_{i,t}$ are average and the standard deviation respectively. If the pixel difference between the current frame and the background is larger than three standard deviations, it will be treated as foreground.

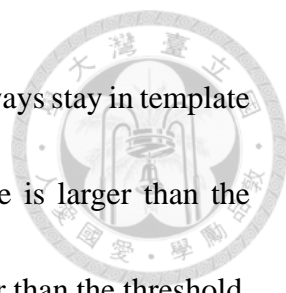
Another improved Gaussian mixture model (Zivkovic, 2004) introduces the learning rate. The learning rate is set to $\alpha = \frac{1}{T}$, a time $T > 1$ during a period of image sequence. This makes the background model able to update in a real time environment. The background forgets the model that has the least weight, it also removes the model if it is outdated. Improved Gaussian model uses the pixel difference as well. The pixel difference will enhance the weight of the possibility of background if it is smaller than three standard deviations. The model in the possibility sequence with the least weight will be removed and replaced with the new one if full. Zivkovic uses clusters to find the background



candidates, and uses maximum likelihood to find the weight of the models. At last, the Dirichlet prior is applied to fulfill the maximum likelihood. Zivkovic also adds a threshold to discard the model if its weight is smaller than the threshold.

2.5.2 Fast Self-Tuning Background Subtraction Algorithm

Fast self-tuning background subtraction algorithm (Bin and Dudek, 2014) uses K templates in the order of the appearance frequency. The 0^{th} template is assumed as the most possible background, descending. In the initialization phase, the algorithm will treat all the objects as foreground. Starts from the 0^{th} template, every pixel will be compared until to the K^{th} template by a background value B and an efficacy counter C . The process orders the templates so that the efficacy decrease from 0 to K . The background value B is defined as the value that has been present at the corresponding pixel for different time scales. Template T_0 always contains the value with the longest time. To adapt to background changes, the new incoming pixel will be classified as a background if the distance between the values B in the corresponding template is smaller than the decision threshold ε . The value B will be updated using a running average. The efficacy of the template that classified the pixel as background will be increase by one, while the efficacies of other templates will be decrease by one. If the efficacy of the template becomes 0, it will become inactive and will be replaced by a new template. The efficacy of the new template will be set to 1.



To ensure the background values with the longest appearance always stay in template 0, a threshold θ_L is used. Only if the efficacy of the first template is larger than the threshold, and the efficacy of the 0^{th} template happens to be smaller than the threshold, the two templates will be swapped. After swapped, the efficacy of the newly updated long-term template will be multiplied to a parameter γ to restrain the generation of ghosts. To prevent a temporarily stay object being insert into the templates, the object must have the efficacy counter C_A greater than the preset threshold θ_A , and has the background value B contained in at least one of its neighbors' background model, judging by the decision threshold ε .

Incoming frames always contain noise. To remove the noise, Wang uses a low resolution and a full resolution image of the same frame. Most noise will be remove in a low-resolution image. If the noise pixel disappears and appears between successive frames, the activity level A of the noise pixel will be increase by one. Once the activity is larger than a threshold β_{TH} , it will be eliminated and treated as background.

Because the decision threshold ε is the most important parameter that directly affects the detection results, Wang uses the activity level A to make the threshold self-adapted. If A is bigger than a threshold β_{INC} , ε increase by δ_{INC} . If A is smaller than a threshold β_{DEC} , ε increase by δ_{DEC} . Using different δ values prevents ε oscillating between two values, and ε is limited between ε_{min} and ε_{max} .



2.5.3 Building PTZ Background Subtraction Model

Using Gaussian Mixture Model to build background is surely a good way to build perfect models. However, the view of the PTZ is not always static. Kang et al. (2003) use an alternative, simple background generation method that requires shorter training time.

Figure 2-2 shows the pseudo code of the fixed-view background model.

```
If background.empty()
{
    Background = incoming_frame;
    Flag_bk(x,y)=0;
}
For all pixels
If  $|I(x,y) - B(x,y)| < \text{Threshold}$ 
    if  $\text{Flag\_bk}(x,y) > 0$ , then  $\text{Flag\_bk}(x,y)--$ 
Else
     $\text{Flag\_bk}(x,y)++$ 
    if  $|I(x,y) - I_{\text{lastframe}}(x,y)| < \text{Threshold}$  and  $|I(x,y) - I_{\text{last2frame}}(x,y)| < \text{Threshold}$ 
        if  $(\text{Flag\_bk}(x,y) > N)$ , then  $\text{Background}(x,y) = I(x,y)$ ,  $\text{Flag\_bk}(x,y) = 0$ 
        else if  $(\text{Flag\_bk} > 0)$ , then  $\text{Flag\_bk}(x,y)--$ 
```

Figure 2-2 Pseudo code of building a PTZ fixed-view background subtraction model by

Kang et al. (2003)

Where I is the incoming frame from the PTZ, Flag_bk is the matrix of consecutive frames. The variable N depends on the frame rate of the video.

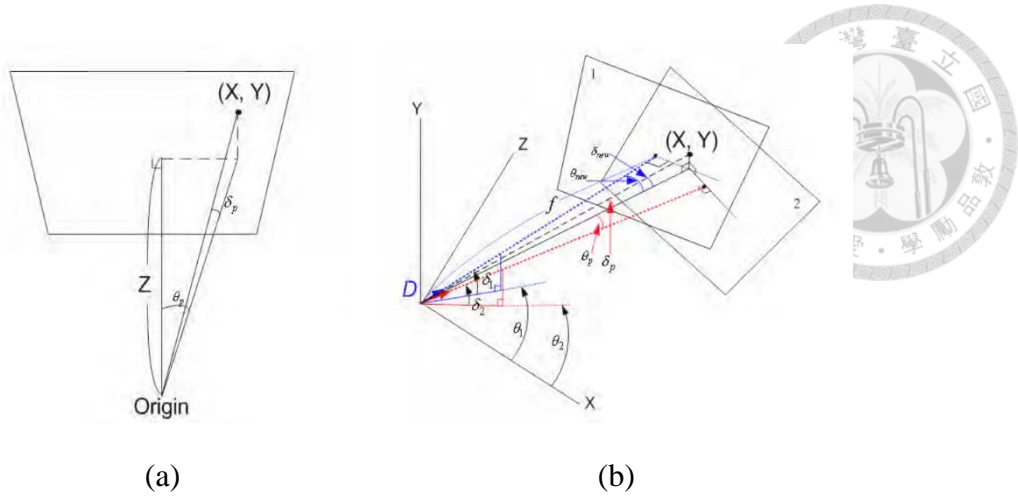


Figure 2-3 Representation of PTZ background remapping. (a)Representing any pixel using pan and tilt for a single image (b)Geometry transformation between different pan-tilt view of PTZ

Next Kang et al. (2003) use a simple geometry transform to transform background to fit different pan-tilt view images. Thus the background can be used even the pan and tilt angle changed. Figure 2-3 shows how this mechanism works and can be written into the following equation,

$$\theta_p = \tan^{-1}\left(\frac{X}{Y}\right), \delta_p = \tan^{-1}\left(\frac{Y}{\sqrt{X^2 + Y^2}}\right)$$

$$Z = \frac{f \cdot \text{width of the PTZ image}}{\text{CCD width}}$$

$$\begin{bmatrix} \cos\delta_{new}\sin\theta_{new} \\ \sin\delta_{new} \\ \cos\delta_{new}\cos\theta_{new} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta_1 & \sin\delta_1 \\ 0 & -\sin\delta_1 & \cos\delta_1 \end{bmatrix} \begin{bmatrix} \cos(\theta_1 - \theta_2) & 0 & \sin(\theta_1 - \theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_1 - \theta_2) & 0 & \cos(\theta_1 - \theta_2) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta_2 & -\sin\delta_2 \\ 0 & \sin\delta_2 & \cos\delta_2 \end{bmatrix} \begin{bmatrix} \cos\delta_p\sin\theta_p \\ \sin\delta_p \\ \cos\delta_p\cos\theta_p \end{bmatrix} \quad (2.2)$$

$$\delta_{new} = \sin^{-1}(\sin\delta_{new}), \theta_{new} = \sin^{-1}\frac{\cos\delta_{new}\sin\theta_{new}}{\cos\delta_{new}}$$

$$X_{new} = Z \cdot \tan\theta_{new}, Y_{new} = \tan\theta_{new} \cdot \sqrt{X_{new}^2 + Z^2}$$

where the X, Y coordinate represent the related coordinate to the center of the PTZ image, θ_p, δ_p are the pan tilt representation for any X, Y pixel of PTZ image, θ_1, δ_1

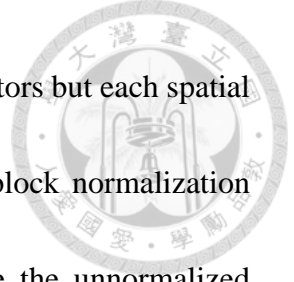
represent the previous pan and tilt angle, θ_2, δ_2 represent the new pan and tilt angle, $\theta_{new}, \delta_{new}$ are the mapped pan and tilt angle for the new image plane.



2.5.4 Histogram of Oriented Gradients for Human Detection

This algorithm is used to calculate the oriented gradients of a small spatial region and is first published by Dalal and Triggs (2005). Every appearance and shape of a small spatial region can be perfectly described with the HOG descriptor. The dense grid of uniformly spaced small spatial regions are called cells. To improve the performance, overlapping local contrast normalization is used to normalize the cells. By deciding the density of the blocks in an image, we can normalize the contrast of the cells located in the blocks. This process makes the feature able to work with less effects from lights and shadows.

When an image is input, the algorithm first normalizes the gamma and the color. After that, the gradients of the image are computed. The orientation of the histogram is voted by the cells. According to the experiment, dividing the orientation into 9 groups out of 360 degrees has the best performance. Next, the cells are gathered into blocks to normalize the contrast. The geometry of the blocks includes rectangle (R-HOG) and circle (C-HOG). Applying the Gaussian Window between them is also needed because somehow it decreases the weight of the edge of an image. The R-HOG blocks works similar to SIFT but computed in dense grids at a single scale without dominant orientation



alignment, The C-HOG blocks work similar to shape context descriptors but each spatial cell contains a stack of gradient-weighted orientation cells. Four block normalization schemes are evaluated for each of the HOG geometry. Let v be the unnormalized descriptor vector, $\|v\|_k$ be its k -norm for $k = 1, 2$, and ϵ be a small constant.

$$\text{L2 - norm, } v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}}$$

L2 - kys, L2 - norm followed by clipping (2.2)

$$\text{L1 - norm, } v \rightarrow \frac{v}{(\|v\|_1 + \epsilon)}$$

$$\text{L1 - sqrt, L1 - norm followed by square root } v \rightarrow \sqrt{\frac{v}{(\|v\|_1 + \epsilon)}}$$

At last, the features are sent into the SVM classifier to detect if the target is human or not. Using a soft ($C=0.01$) linear SVM trained with SVMLight. A Gaussian kernel SVM is also applied to increase performance by about 3 %.

2.6 Object Tracking

When an object is detected, a tracking method is applied to predict the position of the object in the future. A good tracking algorithm is able to detect and predict the object in the interference of noise. The way an algorithm represents an object can be classified into five groups: points, primitive geometric shapes, object silhouette and counter, articulated shape models, and skeletal models. The appearance features of objects can be categorized into 4 groups: probability densities of object appearance, templates, active



appearance models, and Multiview appearance models (Yilmaz et al., 2006). The common algorithms are as follows.

2.6.1 Bayesian Filtering

Bayesian filtering is relevant to density functions, which are approximated by kernel-based representations and propagated over time (Bohyung et al., 2005). The process and measurement model are given by

$$x_t = g(x_{t-1}, u_t) \quad (2.3)$$

$$z_t = h(x_t, v_t) \quad (2.4)$$

where u_t and v_t are process and the measurement noise. The state variable x_t characterized by its probability density function (PDF) is estimated using the value at $t - 1$, and is corrected afterwards by the measurement z_t . The filtering contains two stages: status predict, and status update. The stages can be represented by the following equations.

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \quad (2.5)$$

$$p(x_t|z_{1:t}) = \frac{1}{k} p(x_t|x_t)p(x_t|z_{1:t-1}) \quad (2.6)$$

The most common PDF used in state variable is the Gaussian model, as equation 2..

P_t^i is the corresponding covariance matrices at time step t , and κ_t^i is the Gaussian weight with the total sum equals to 1.

$$p(x_{t-1}|z_{1:t-1}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\kappa_{t-1}^i}{|P_{t-1}^i|^{1/2}} \exp\left(-\frac{1}{2} D^2(x_{t-1}, x_{t-1}^i, P_{t-1}^i)\right) \quad (2.7)$$



2.6.2 Mean-Shift Object Tracking

The mean-shift tracking framework consists of target representation and target localization (Comaniciu et al., 2003). In target representation, a reference target model is chosen; we will have to characterize the target by a feature space. The reference target model is represented by its probability density function (PDF) q in the feature space. In the subsequent frame, a target candidate is defined at location y , and is also characterized by the PDF. To make the computation cost reach the real-time processing, m-bin histogram is used instead. The target model and candidate is as following equations.

$$\text{target model:} \quad \hat{q} = \{\hat{q}_u\}_{u=1\dots m} \sum_{u=1}^m \hat{q}_u = 1 \quad (2.8)$$

$$\text{target candidate:} \quad \hat{p}(y) = \{\hat{p}_u(y)\}_{u=1\dots m} \sum_{u=1}^m \hat{p}_u = 1 \quad (2.9)$$

The target model is represented by an ellipsoidal region in the image. All targets are first normalized to a unit circle to eliminate the influence in different dimensions. The target model is then computed using the multivariate kernel density estimation method as equation

$$\hat{q}_u = \frac{1}{\sum_{i=1}^n k(\|x_i\|^2)} \sum_{i=1}^n k(\|x_i\|^2) \delta[b(x_i) - u], \quad (2.10)$$

where δ is the Kronecker delta function, $k(x)$ is the kernel profile, and h is the bandwidth. The target candidate is also computed using the same kernel profile, but with different bandwidth, as equation 2.11.

$$\hat{p}_u = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i) - u], \quad (2.11)$$

However, representing by color info isn't robust enough. Comaniciu et al. (2003) mask the target with an isotropic kernel in the spatial domain $\hat{\rho}(y)$, with pixels weighted similar to the desired color. The presence of the objects in the second frame can be represented by equation 2.12.

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] \quad (2.12)$$

In target localization phase, the model chosen in the current frame will be used the search in the model's neighborhood in the next frame. The best candidate is found by maximizing the similarity function $\hat{\rho}(y)$, or minimized the Bhattacharya distance of the two discrete distribution. The similarity function can be further computed using the Taylor expansion, as equation 2.13.

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \quad (2.13)$$

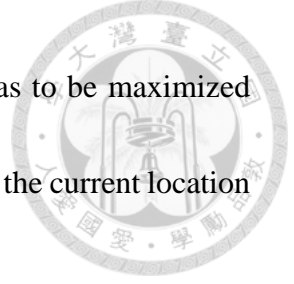
Recalling equation 2.11 we can get

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_0) \hat{q}_u} + \frac{c_h}{2} \sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right), \quad (2.14)$$

where

$$c_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \quad (2.15)$$

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(x_i) - u] \quad (2.16)$$



To minimize the Bhattacharyya distance, the second term in 2.14 has to be maximized using the mean-shift algorithm. The kernel is recursively moved from the current location to the new location according to the following equation

$$\hat{y}_i = \frac{\sum_{i=1}^{n_h} x_i w_i g(\|\frac{\hat{y}_0 - x_i}{h}\|^2)}{\sum_{i=1}^{n_h} w_i g(\|\frac{\hat{y}_0 - x_i}{h}\|^2)}, \quad (2.17)$$

where $g(x) = -k'(x)$. A threshold ε is used as the error of the Bhattacharyya distance.

The searching will stop if $\|\hat{y}_1 - \hat{y}_0\| < \varepsilon$.

2.6.3 Kalman Filtering

Kalman filter (Kalman, 1960) is an optimal estimator, and its known as the best linear estimator. Its model is a Bayesian model similar to a hidden Markov model (HMM)(Rabiner, 1989). The noise is assumed Gaussian (Diebold, 2007). The Kalman filter estimates the current state variables with the previous positions, speed, and other uncertainties. The measurement state is then feedback to the controlled system. The following equations shows how Kalman filter works

$$x(k) = T(k)x(k-1) + M(k)u(k) + v(k) \quad (2.18)$$

$$z(k) = H(k)x(k) + w(k), \quad (2.19)$$

where $T(k)$ is the transition matrix of the state vector $x(k)$, $M(k)$ is the effect of input control working on the control variables $u(k)$, $H(k)$ is the observation model of the measurement state $z(k)$, $v(k)$, $w(k)$ are the process noise and the measurement noise



respectively. The matrix of quadratic form defining error criterion $Q(k)$ and $R(k)$ are assumed the average of zero.

2.6.4 Particle Filtering

Particle filter (Arulampalam et al., 2002) or sequential Monte Carlo method (SMC)(Doucet et al., 2001) are one of the method to solve non-linear and non-Gaussian problems. It is based on point mass representations of probability densities and can be applied to any state-space model. Because most problems are time dependent and contains noise, we will have to build an estimate model in a discrete-time and state-space form. To define the problem of tracking, we can use the following equations to represent the state sequence and the measurement of a target

$$x_k = f_k(x_{k-1}, v_{k-1}) \quad (2.20)$$

$$z_k = h(x_k, w_k), \quad (2.21)$$

where f estimates the current state according to the previous state, h is a possibly non-linear function. v , w are the process noise and the measurement noise respectively.

The sequential important sampling (SIS) algorithm based on SMC method is a generic framework of the particle filter. The SMC method represents the posterior density function by a set of random samples with associated weights. The result of the SMC method is more likely to be an experimental result rather than a calculation result. If the number of samples becomes very large, the result will become an equivalent



representation to the usual functional description of the pdf and approaches the optimal Bayesian estimate. We then describe the desired non-linear tracking system in the form of posterior pdf p

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(x_k - x_k^i), \quad (2.22)$$

where z is the measurement state. The system is described at the set of all states up to time k using N_s of particle x_k^i to represents the probability distribution in the state of x_k . Every state x_k^i is associated with a weight w_k^i

$$w_k^i \propto w_{k-1}^i \frac{p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)}, \quad (2.23)$$

where q is the important density function of the distribution function of the sampled particles. If we choose the importance density function to be the prior,

$$q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i), \quad (2.24)$$

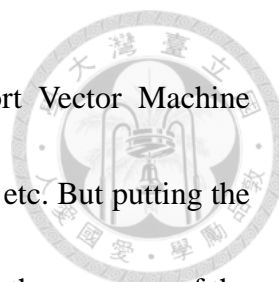
and equation 2.23 can be simplified into

$$w_k^i \propto w_{k-1}^i p(z_k | x_k^i). \quad (2.25)$$

Any state x_k^i and its probability in any time can be calculated by the SIS algorithm.

2.7 Behavior Recognition

Rule-based behavior recognition methods are limited to predefined anomalies. Thus, the researches have been a paradigm shift from rule-based to statistical-based methods (Popoola and Kejun, 2012). The statistical methods can be fulfill using learning algorithms, such as Hidden Markov Model (HMM)(Zhang et al., 2005), Bayesian model



(Boiman and Irani, 2007), clustering (Zhou et al., 2007), Support Vector Machine (SVM)(Wu et al., 2005), neural network (Foroughi et al., 2008), and etc. But putting the raw data from the image sequence into the models directly may lower the accuracy of the recognition results, some pre-processing must be done to reduce the noise of the raw data. We called these pre-processing data features, and these features will be later used for training and recognizing. There are lots of feature extraction methods, for example, object trajectory (Wiliem et al., 2008), histogram related features (Tao and Shaogang, 2008), optical flow (Feizi et al., 2013), shape related algorithms, and etc.

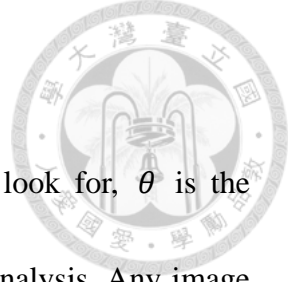
Some pre-processing technics even use one or more methods. Chang et al. (2013) use bag-of features and skeleton graph to recognize the behavior. To recognize the human behavior, two model is used. First, a shape representation method is applied by using bag-of-features to approach. Then these models are insert into the second model. The motion then is extracted by skeleton graph, and is then saved into codebook. A SVM is used to train these codebook-features.

2.7.1 Gabor Filtering

Frequency and orientation of Gabor filters (Feichtinger and Strohmer, 1998) are similar to human visual system. They are also useful to extract features from an image.

The following equations are the two-dimensional Gabor filters in discrete domain.

$$G_1(i, j) = C_1 e^{-\frac{i^2+j^2}{2\sigma^2}} \cos(2\pi f(i\cos\theta + j\sin\theta)) \quad (2.26)$$



$$G_2(i, j) = C_2 e^{-\frac{i^2+j^2}{2\sigma^2}} \sin(2\pi f(i\cos\theta + j\sin\theta))$$

where C_1, C_2 are the normalizing factors, f is the frequency to look for, θ is the texture orientation angle, σ is the size of the image region being analysis. Any image applied with the Gabor filter with different frequencies and orientations is able to produce larger variance features and makes the classifiers easier to find a solution.

2.7.2 Least Squares Support Vector Machine

Support Vector Machine (SVM)(Vapnik, 1995) is usually used in machine learning, and it's one of a supervised learning method. The main purpose of the SVM is classification and regression. The main purpose of the support vector is to find the maximum margin in a training data set that gives the most information points in classification. Assumed a given training set of N data points $\{y_k, x_k\}_{k=1}^N$ as follows:

$$y(x) = \text{sign} \left[\sum_{k=1}^N a_k y_k \psi(x, x_k) + b \right] \quad (2.27)$$

where $x_k, y_k \in \mathbb{R}$ are the kth input and output pattern respectively. For $\psi(x, x_k)$ one has the following choices according to the different types of SVM used. The SVM can classify using linear and non-linear at the same time. The raw data is transformed to a higher (Vapnik-Chervonenkis, VC) dimension, and this helps the algorithm to find a hyperline that has the maximum margin. A risk minimization principle is used to minimize the risk bound as equation 2.28.



$$\min_{\omega, \xi_k} \mathfrak{J}(\omega, \xi_k) = \frac{1}{2} \omega^T \omega + c \sum_{k=1}^N \xi_k \quad (2.28)$$

where ξ_k is a margin related variable. The solution is given by the saddle point of Lagrangian, and applying the computed results to equation 2.28 leads us to the following quadratic programming problem.

$$\max_{a_k} Q(a_k; \varphi(x_k)) = -\frac{1}{2} \sum_{k,l=1}^N y_k y_l \varphi(x_k)^T \varphi(x_l) a_k a_l + \sum_{k=1}^N a_k \quad (2.29)$$

Such that

$$\sum_{k=1}^N a_k y_k = 0, 0 \leq a_k \leq c, k = 1, \dots, N \quad (2.30)$$

where φ is a nonlinear function which maps the input space into higher dimension. The function $\varphi(x_k)$ in equation 2.29 is related to $\psi(x, x_k)$ by imposing

$$\varphi(x)^T \varphi(x_k) = \psi(x, x_k). \quad (2.31)$$

The classifier then become

$$\max_{a_k} Q(a_k; \psi(x, x_k)) = -\frac{1}{2} \sum_{k,l=1}^N y_k y_l \psi(x, x_k) a_k a_l + \sum_{k=1}^N a_k \quad (2.32)$$

Because the matrix associated with equation 2.32 is not indefinite, the solution of 2.32 will be global. The selected SVM with minimal VC dimension can be found by solving equation 2.32 and computing the ball containing the points $\varphi(x_1), \dots, \varphi(x_N)$ using the Lagrangian.

Least Square SVM is (LS-SVM) is least square versions of SVM (Suykens and



Vandewalle, 1999). Instead of the quadratic programming, LS-SVM finds the classifier by solving the linear set of equations 2.33-2.34 (Fletcher, 1987).

$$\begin{bmatrix} 0 & -Y^T \\ Y & ZZ^T + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{1} \end{bmatrix} \quad (2.33)$$

where $Z = [\varphi(x_1)^T y_1; \dots; \varphi(x_n)^T y_N]$, $Y = [y_1; \dots; y_N]$, $\bar{1} = [1; \dots; 1]$, $e = [e_1; \dots; e_N]$,

$\alpha = [\alpha_1; \dots; \alpha_N]$. Mercer's condition can be applied to the matrix $\Omega = ZZ^T$

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(x_k)^T \varphi(x_l) \\ &= y_k y_l \psi(x_k, x_l) \end{aligned} \quad (2.34)$$

The least square SVM has a better generalization performance and lower computational cost compare to the original SVM.

2.7.3 Neural Network

Neural Network (NN)(Aleksander and Morton, 1995) is consist of neutral nodes.

Each node accepts several input function values with different weight. These values will be sum up and send into a function to produce calculation result. Every node can accept multiple inputs from same layer and send the calculation result to other nodes in same layer.

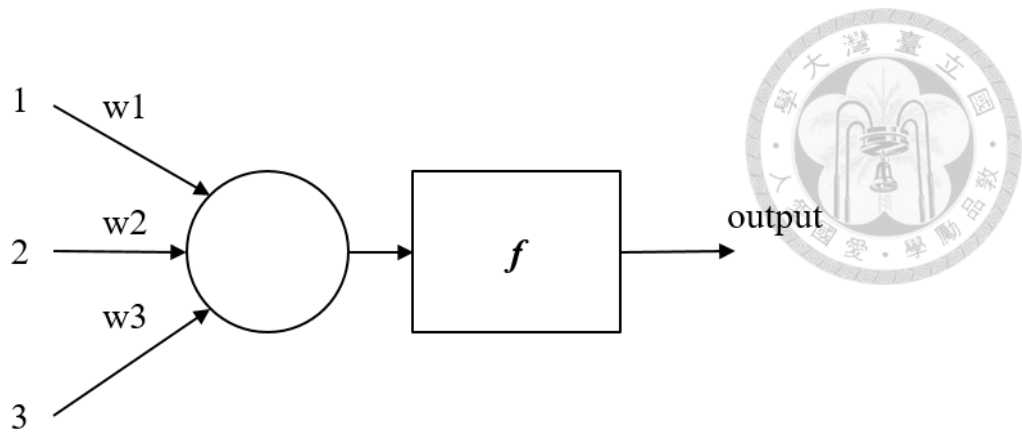


Figure 2-4 One node in a Neural Network

Each Neural Network has one or more layers to connect between the nodes. One layer NN uses the layer as input also as the output. Two layers NN has a layer of input and a layer of output. Three or more layers NN has a layer of input at where data input and a layer of output at output, and the rest of the layers are the hidden layers. Because the output of one NN layer is a vector, the size of the vector is the same as the number of the nodes for any single layer.

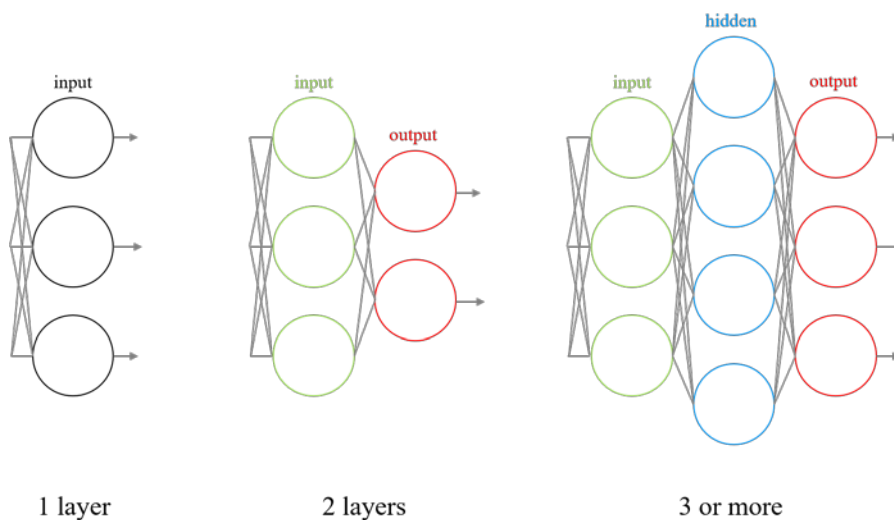
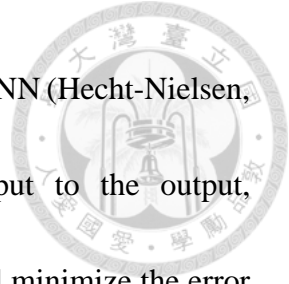


Figure 2-5 Different layers of Neural Network



Another popular type of Neural Network is the backpropagation NN (Hecht-Nielsen, 1989). Instead of finding and tuning the values from the input to the output, backpropagation starts from the output back to the input. It finds and minimize the error produced during each layer.

2.7.4 K-Nearest Neighbor

K-nearest Neighbor (Cover and Hart, 1967) is a non-parametric method used in classification. The classification of a point is decided by voting method of its neighbor. The value k , positive number, usually small, decides how many neighbors involved the voting process. If $k = 1$ then the classification result will be decided by the nearest point of the new point. It is regarded as the simplest method of machine learning. During a voting process, the decision weight is opposite to the distance, usually the Euclidean distance, that is, the bigger the distance, the smaller the weight. This method also has drawbacks. If one sample appears too often, it will directly affect the result of the classification. Determine a good value k will lower the effect of noise and clarify the edges between groups.

2.7.5 Naive Bayes

Naïve Bayes (Russell and Norvig, 2009) is based on applying Bayes' theorem. We assume between all the random variables are independent (Eq. 2.35). The denominator of equation 2.36 is independent of C and for each of the value x_i is given, we then can



modify the Bayes theorem into equation 2.37 by chain rule. 2.392.38

$$P(C|x) = P(x_1|C)P(x_2|C) \dots P(x_n|C) \quad (2.35)$$

$$P(C|x_1 \dots x_n) = \frac{P(x_1 \dots x_n|C)P(C)}{P(x_1 \dots x_n)} \quad (2.36)$$

$$P(C|x_1 \dots x_n) \propto P(C)P(x_1|C)P(x_2|C, x_1) \dots P(x_n|C, x_1 \dots x_{n-1}) \quad (2.37)$$

We assume for every x_i compare to other features x_j ($j \neq i$) is independent, we can modify 2.37 by using equation 2.38, and can be represents by equation 2.39.

$$P(x_i|C, x_j) = P(x_i|C) \quad (2.38)$$

$$P(C|x_1 \dots x_n) \propto \prod_{i=1}^n P(x_i|C) \quad (2.39)$$

By using this classifier, we can calculate the probability of the next input data. The result will be predicted by comparing the probabilities of each group. With the combination of Expectation-maximization algorithm, we can iterate continually to find out the unknown variables until converged. The classifier for Naïve Bayes can be represented as the following equation.

$$classifier(x_1, \dots, x_n) = \underset{k \in \{1 \dots K\}}{\operatorname{argmax}} P(C_k) \prod_{i=1}^n P(x_i|C_k) \quad (2.40)$$

Chapter 3 Materials and Methods



3.1 System Architecture

3.1.1 Hardware Design

The cooperative system is separated into two parts: the static camera system and the pan-tilt-zoom (PTZ) camera. We use a Sony Network Camera (SNC) EB-630 (Figure 3-1 a) plus a Theia SY125M Ultra Wide Lens (Figure 3-1 b) as our static camera. We use ACTi® CAM-6630 High Speed Dome Camera (Figure 3-1 c) as the PTZ camera. Both cameras provide their own Software Development Kit (SDK). We use HyperText Transfer Protocol (HTTP) and Common Gateway Interface (CGI) to control the Sony camera, including setting the exposure time, the frame rate, the shutter speed, the wide-dynamic range function, and etc. We use HTTP and RS232 to control the PTZ camera. We can enter the absolute position to make the PTZ itself to turn around in two angle: horizontal and vertical. We are also able to control the focal length, and the zooming rate to look at the object closer. The specification of each hardware is shown in Table 3-1 to Table 3-3.

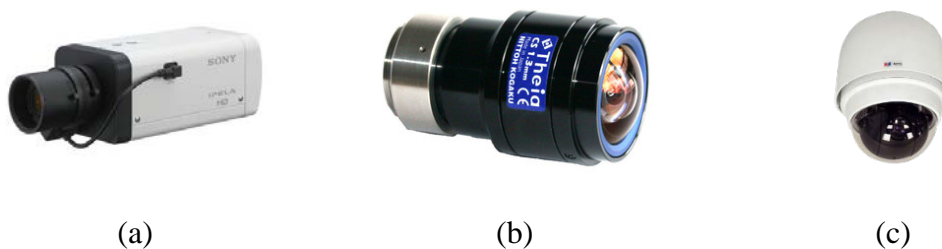
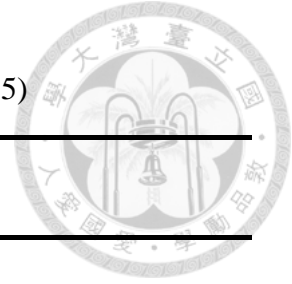


Figure 3-1 Picture of the Chosen Hardware (a) SNC EB-630 (Sony, 2015) (b) Theia

SY125M (Theia, 2015) (c) ACTi® CAM-6630 High Speed Dome Camera (ACTi, 2015)

Table 3-1 Specification of SNC EB-630 (Sony, 2015)



Function	Specification
Resolution	1920x1080 pixels
Focal Length	2.8 to 8.0 mm
Frame Per Second (FPS)	30 fps in full HD resolution
Zoom Ratio	Optical zoom 2.9x / Digital zoom 4x Total zoom 11.6x
Shutter Speed	1 / 1 to 1 / 10,000 s
S / N Ratio (Gain 0 dB)	More than 50 dB
Gain	Auto
Dynamic Range	Equivalent to 90 dB with View-DR technology
Day / Night	True D / N (Infrared light)
Lens	CS-mount varifocal lens
White Balance	ATW, ATW-PRO, Fluorescent lamp, Mercury lamp, Sodium vapour lamp, Metal halide lamp, White LED, One push WB, Manual

Table 3-2 Specification of Theia SY125M (Theia, 2015)

Function	Specification
Iris Type	Manual Iris
Focal Length	1.3mm using Linear Optical Technology®
F / #	F / 1.8 to closed
Resolution	Up to 5 megapixels
IR Correction	No (IR transmitting but not D /N corrected)
Mount Type	CS-mount



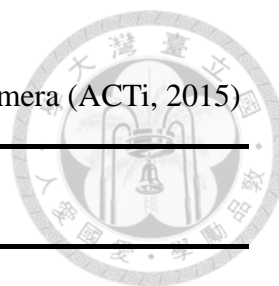


Table 3-3 Specification of ACTi® CAM-6630 High Speed Dome Camera (ACTi, 2015)

Function	Specification
Resolution	720x480 pixels
Focal Length / CCD size	3.4~119 mm / 3.2 x 2.4 mm
Frame Per Second (FPS)	30 fps at full D1 resolution
Zooming Rate	27x optical
Zooming Speed	4.6 sec (1x to 35x)
Horizontal Spinning Angle	0~360 degrees
Vertical Spinning Angle	-10~100 degrees
Horizontal Moving Speed	1~90 degrees / s
Vertical Moving Speed	1~90 degrees / s
Absolute Position	Yes

3.1.2 Software Design

The software is written in C++ language by Qt (Digia, 2015). The software is a user-friendly design. The user interface is can be preview in Figure 3-2 and Figure 3-32. The main cooperative surveillance system (Figure 3-2) has the ability to control the PTZ manually by mouse. The size of the main screen and the child screen can also be zoomed or dragged around with the mouse and provide a Video Live View of both cameras. The



Object View Manager (Figure 3-32) provides a friendly GUI for users to look up every tracked object detail including the images of the both cameras after the final analysis.

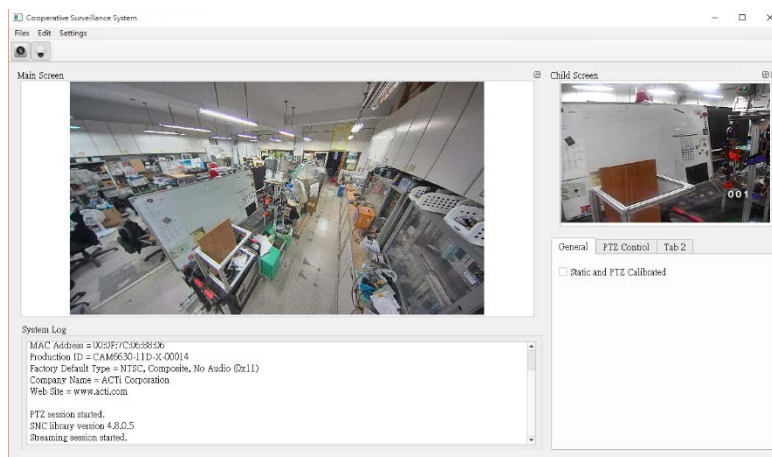


Figure 3-2 Software GUI design of the Cooperative Surveillance System

3.1.3 Data Structure

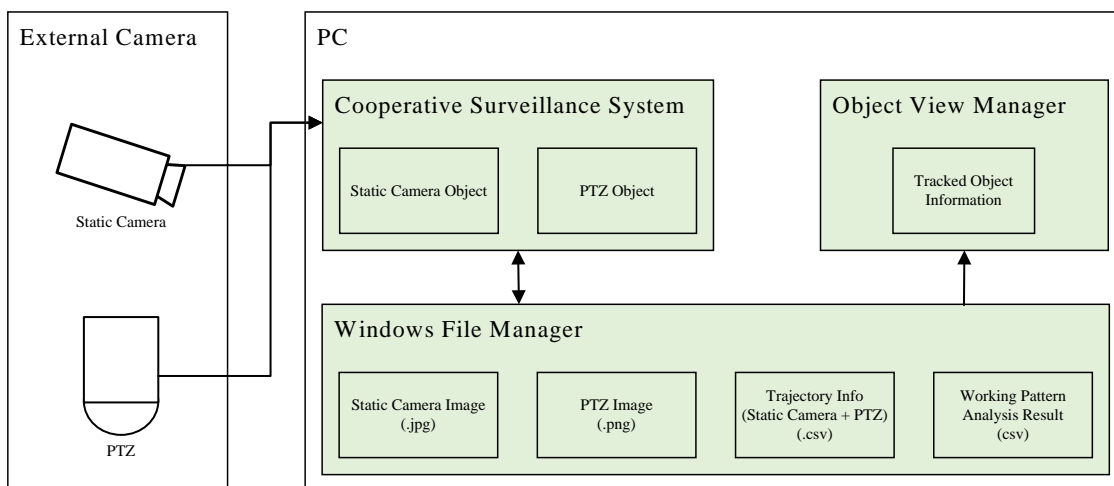
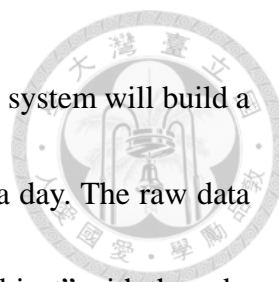


Figure 3-3 Data Structure Diagram of Our Cooperative Surveillance System

We design our data structure to record every possible information we need. Figure 3-3 shows the diagram of our system. We use the Windows native file manager to store our files, including images, trajectory text files instead of building our own database. The



internal category functions from Windows is enough for our use. The system will build a category in the format of csv file for the images and text files once a day. The raw data structure in the Cooperative Surveillance system are in the unit of “Object” with the sub-information such as time, duration, trajectory, sub-image, etc. Table 3-4 shows the data structure of our system. As for the data structure in the Object View Manager, it has a unit of “Trajectory Object Information”, which includes all the information from the Windows File Manager. The more rules and modules are used in the working pattern analysis, the more complicated the structure is.

Table 3-4 Data Structure of the Cooperative Surveillance System

Static Camera	PTZ
Object	Object
● Time	● Time
● Duration	● Close-up images
● Trajectory	
● Foreground images of the object for each recorded coordinate	
● Behavior	

3.2 Cooperative system

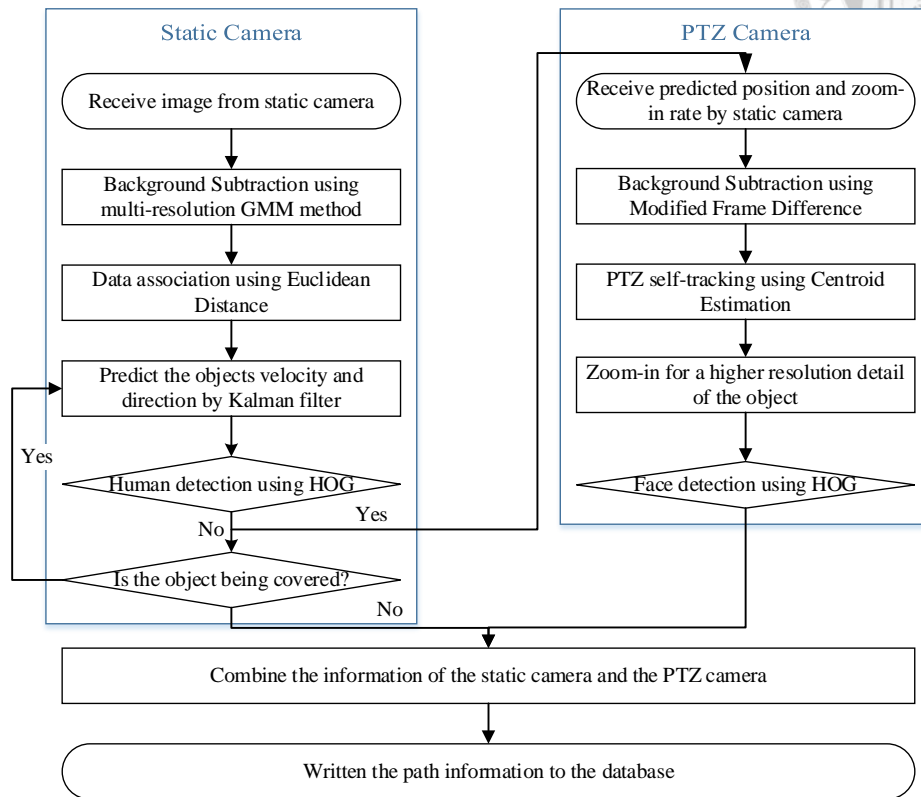


Figure 3-4 Diagram of Network Control Center

The main goal of the cooperative surveillance system is to track and record the trajectories of the human activities. We construct a Network Control Center (NCC) to let cameras cooperate and trade information to track the objects. Different cameras have different advantages of acquiring unique information. The static camera collects the wide-angle view scenes, while the PTZ camera collects the close-up and high-resolution image of any tracked object. The main searching and tracking algorithm is processed by the image from the static camera. Foreground information is used to associate the same object

between frames. The algorithms and the strategies used in the system will be introduced in the next section.



3.3 Object Detection and Tracking

3.3.1 Gaussian Filter and Interactive Multiple Model (IMM) Filter

We use simple image processing method to extract the foreground objects from the background by multi-resolution Gaussian Mixture Model (GMM) background subtraction method (Zivkovic, 2004). Noise is a non-constant information based on time, resampling images in different resolutions eliminates the noise in a video. Once the noise is removed, the contour of the foreground object can be calculated. The contour information calculated in the new frame is used to identify the tracked object between frames. If the Euclidean distance between the new object and the tracked object is less than a calculated threshold (equation 3.2), they are considered as the same object. If there are multiple choices, the system chooses the most possible location by estimating the tracked objects future position with its velocity and direction. The threshold differentiates along the y-axis. Due to our settings of our system, the object is much near to the camera lens compare to the old system. If the object is close to the lens, the proportion of the object on the image will be bigger. Bigger Euclidean threshold is required to prevent the loss of the closer object. The threshold is determined according to the distance from the object to the camera. We assume one-step of a human is roughly 0.8 m. By the pitch of



the camera, we can calculate the corresponding pixel length of a step at $y = h/2$ with the following formula:

$$\frac{\text{one step pixel length}}{\frac{x}{2}} = \frac{0.8}{\frac{L}{\cos(\theta)} \cdot \tan(\frac{\varphi}{2})} \quad (3.1)$$

where L = the distance from the camera lens to the ground point where the camera looks at, θ = the pitch of the camera, φ = the FOV of the camera, and x = the x resolution of the image.

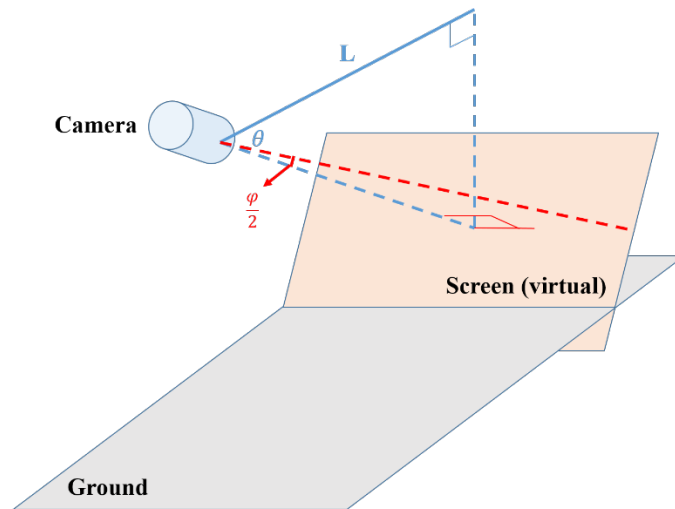


Figure 3-5 Illustration of Dynamic Euclidean Threshold (equation 3.1, 3.2)

After finding the vanish point of the y axis, the threshold ($T_{E.D.}$) for any new object then can be calculated by the following equation,

$$T_{E.D.} = c \cdot (\text{one step pixel length} - T_{init}) \cdot \frac{y_o - Y_{v.p.}}{\frac{h}{2} - Y_{v.p.}} + T_{init} \quad (3.2)$$

where c = fine tuning constant, y_o = the y position of a detect object, and $Y_{v.p.}$ = y axis of the vanishing point. The T_{init} is the initial threshold at the vanishing point of the



image, and can be calculated using similar equation as equation 3.1. Just change the pitch of the camera to the angle of the vanishing point to the horizontal line.

```
//Data association
for(int i = 0; i <= obj.size(); i++)
{
    If EuclideanDistance(obj[i], obj_last_frame[i]) < dynamicThreshold
        connect(obj[i], obj_last_frame[i])
    Else if obj[i].gabor() is similar to obj_last_frame[i].gabor()
        connect(obj[i], obj_last_frame[i])
}
//Check association result
for(int i = 0; i < connected_obj.size(); i++)
{
    EstimatePosition = IMM & Kalman(obj_last_frame[i])
    If EstimatePosition is not similar to obj[i]
        disconnect(obj[i], obj_last_frame[i])
}
}
```

Figure 3-6 Pseudocode of Data Association and Correction

After the correcting with the previous location(s) and the present location, we are able to predict the object velocity and the direction of the targeted object by Kalman filter (Kalman, 1960) and Interactive Multiple Model (IMM) filter (Challa et al., 2011). This not only tells the PTZ to capture the object easier, but also tells the system to know which direction the object is moving when the object is covered. We can also check if the linking is correct if the connected location is unexpectedly. In IMM filter, we choose a constant velocity (CV) model and a constant acceleration (CA) model. We assume the motion model of an object has different probabilities to keep in the same model for the next step.

The probabilities are determined by observing them in the testing environment. We then estimate the object states by these two motion models (CV and CA) respectively, and get the final state by the weighted sum calculated by the likelihood function. The predicted object state can be estimated precisely and is ready to be sent to the PTZ camera to capture the object for more details.

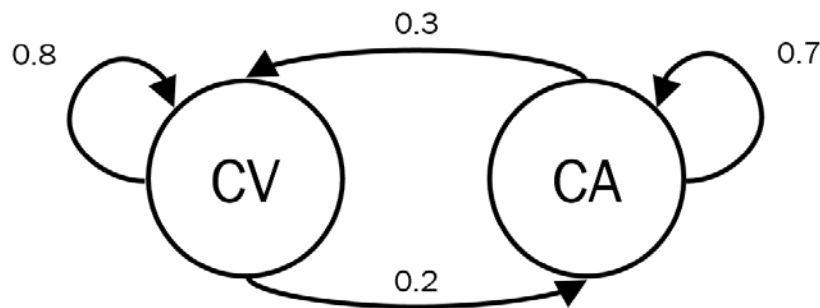


Figure 3-7 IMM model for Network Control Center (NCC)

3.3.2 Geometry Transform between Cameras

The geometry location estimated by the Kalman filter and IMM filter belongs to the coordinate of the static camera. In order to change these to the PTZ coordinates, we use a simple geometry transform according to the formula below. We first change the static camera to the world coordinate, then we change the world coordinate to the PTZ coordinate (Fig. 2).



$$\theta_s = \frac{1}{s}x' + b_1, \quad x' = x_s \cos(\theta) - y_s \sin(\theta)$$

$$\phi_s = \frac{1}{s}y' + b_2, \quad y' = x_s \sin(\theta) + y_s \cos(\theta)$$

$$X_T = Z_S \cot(\phi_s) \cos(\theta_s) + X_S$$

(3.3)

$$Y_T = Z_S \cot(\phi_s) \sin(\theta_s) + Y_S$$

$$\theta_p = \tan^{-1}\left(\frac{Y_T}{X_T}\right)$$

$$\phi_p = \tan^{-1}\left(\frac{Z_P}{\sqrt{X_T^2 + Y_T^2}}\right) + \xi_p$$

where x_s, y_s = the position of the object in the static camera coordinate, θ_s, ϕ_s = calculated pitch and raw of the object in the static camera coordinate, $1/s, b_1, b_2$ = constant parameters, X_T, Y_T = the position of the object in the world coordinate, θ_p, ϕ_p = the pan and tilt of the object in the PTZ coordinate.

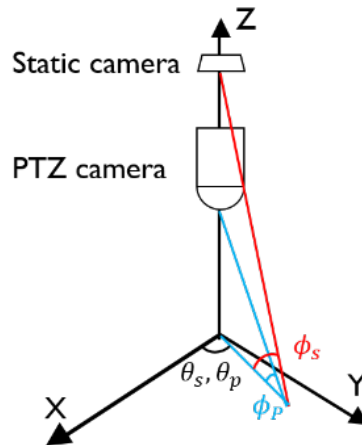
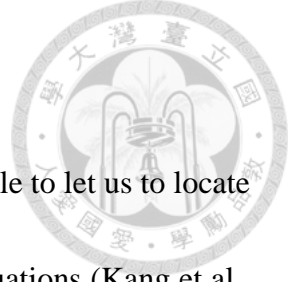


Figure 3-8 Illustration of Geometry Transform



3.3.3 PTZ Image Mapping to PTZ Sphere coordinate

Mapping the PTZ image back to its own sphere coordinate is able to let us to locate the detailed object more precisely. We use simple transformation equations (Kang et al., 2003; 余, 2012) to find the estimated pan, tilt angles from a known x, y coordinate from the image as shown in equation 3.4,

$$\begin{aligned}
 Z &= \frac{(focal \cdot zooming \ rate) \cdot pixel \ width \ of \ PTZ \ image}{CCD \ width} \\
 a &= \frac{Z}{\tan(180^\circ - \phi)}, b = \frac{a}{\sin(180^\circ - \phi)} \\
 t &= \tan^{-1} \frac{a \cdot (x - \frac{w}{2})}{b \cdot (y - \frac{h}{2} + a)}, A = \frac{(y - \frac{h}{2}) + a}{\cos(t) + 0.000001} \\
 d\theta &= \tan^{-1} \left(\frac{(x - \frac{w}{2})}{Z \cdot \cos(\phi)} \right) \\
 d\phi &= \tan^{-1} \left(\frac{A - a}{Z} \right) \\
 \theta_{estimated} &= \theta + d\theta \\
 \phi_{estimated} &= \phi + d\phi
 \end{aligned} \tag{3.4}$$

where θ, ϕ is the pan and tilt angle of the PTZ, x, y is the x, y coordinate of the PTZ image, w, h is the pixel width and the pixel height of the PTZ image. The new estimated pan, tilt angles are calculated by finding the relationship from the old pan, tilt angles.

By using the raw CCD width and the focal of the PTZ camera, we are able to calculate the Z value in any zooming rate. The view of the PTZ is a south hemisphere, so we minus 180 degree by tilt to make the warping direction upside down.

3.3.4 PTZ Self-Tracking Algorithm



Background Subtraction using Frame Difference

The estimated coordinate from the geometry transformation may be incorrect due to the errors or time lagged. After the PTZ received the coordinate, we design a mechanism for the PTZ to relocate the position of the tracked object. The background subtraction model is built with the pseudo code shown in Figure 2-2.

We modified the value $N = 2$ to make the model update very fast. The goal of the self-tracking is to relocate the found object to the center. Changing the N to a very small value actually makes the system more like frame difference than building a background model, but with less noise and more tolerance. The geometry transform (equation 2.2) between different pan-tilt angle views is also used to prevent rebuilding the background model repeatedly. The foreground image is processed with simple dilate and erode to eliminate noise.

Tracked object Centroid Estimation

After the foreground is extract from the PTZ image, we will try to find the object that needs to relocate. The predicted coordinate from the static camera tells the PTZ to move to an estimated location. The tracked object must be near the center to the PTZ image in theory. We add a weighting to find the centroid of all the foreground pixels using Gaussian function. The pseudo code is shown in Figure 3-9.



For every foreground pixel

Centroid_x(x, y) += x * PixelIntensity(x, y) * Gaussian (|w/2-x|)

Total_x += PixelIntensity(x, y) * Gaussian (|w/2-x|)

Centroid_y(x, y) += y * PixelIntensity(x, y) * Gaussian(|h/2-y|)

Total_y += PixelIntensity(x, y) * Gaussian(|h/2-y|)

Centroid_x /= Total_x

Centroid_y /= Total_y

Figure 3-9 Pseudo code of finding the centroid of the tracked object

Figure 3-10 is the visualize example how self-tracking works. In Figure 3-10 (a) the two man on the left side screen is moving, so they are captured by the PTZ. The PTZ then try to move to the left a little bit. However, in Figure 3-10 (b) the man with black shirt leaves causing the two moving objects separated, the PTZ then changed the center to capture both moving objects. At last, the only moving person is the one with gray shirt (Figure 3-10 (c)). The PTZ then move the center back to the man with gray shirt.

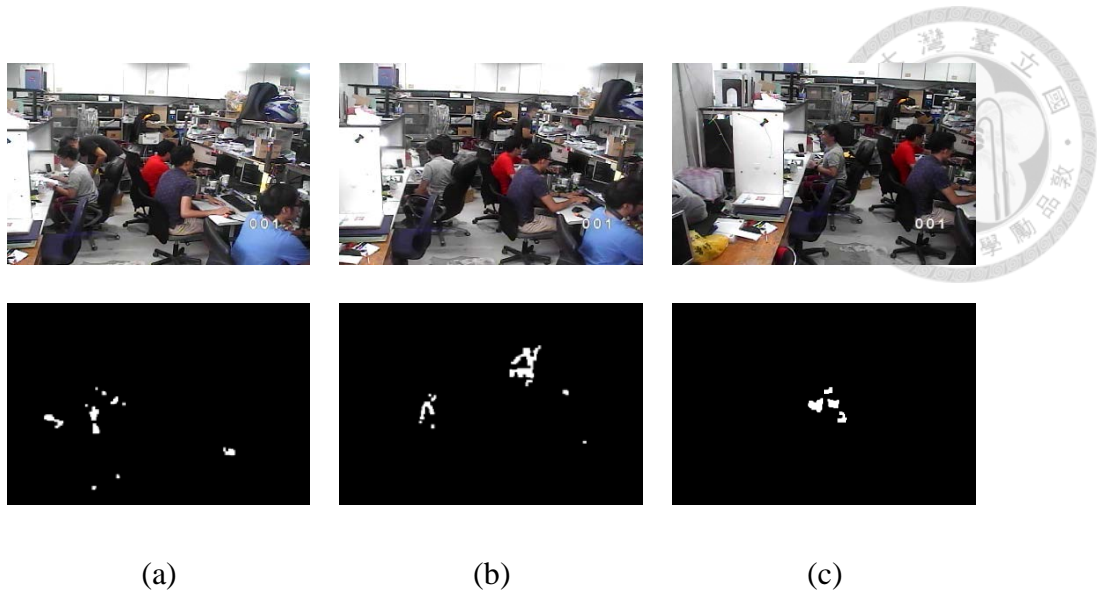


Figure 3-10 Centroid estimation process of PTZ self-tracking. (a) is the initial position received from static camera. (b) is the computed centroid view. (c) is a recomputed centroid view after the foreground object changed. Note that only moving objects will be captured on the screen.

Zooming Rate Estimation

Finally, the size from the foreground pixels will determine the zooming rate of the PTZ. The zooming rate is calculated by the following equation.

$$\begin{aligned}
 dx &= foreground(x_{top} - x_{bottom}) \\
 dy &= foreground(y_{top} - y_{bottom})
 \end{aligned}
 \tag{3.5}$$

$$f(\text{zooming rate}) = \begin{cases} \frac{w}{dx}, \frac{w}{dy} < \frac{h}{dy} \\ \frac{h}{dy}, \frac{w}{dx} > \frac{h}{dy} \end{cases}$$

PTZ Tracking Strategy

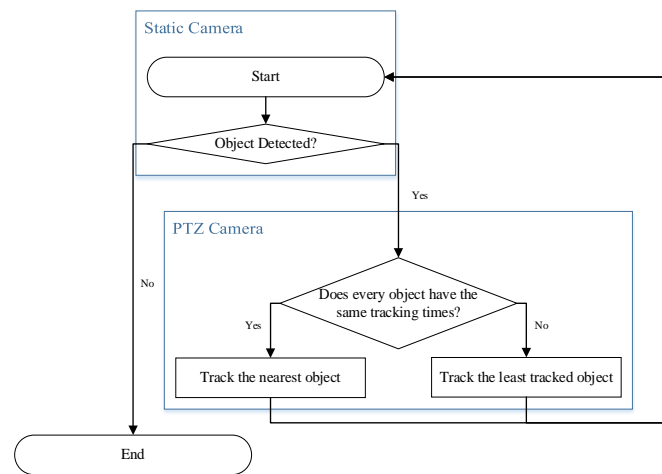


Figure 3-11 The PTZ tracking strategy of the previous system

Figure 3-11 shows the PTZ control strategy of the previous system (余, 2012). This efficient tracking strategy is able to collect information with minimum moving distance and minimum reduced time. However, due to the previous reason told in the beginning of this section, we modified the strategy a bit to gather the information more completely. The PTZ not only received the estimated position from the static camera but also able to track object on its own.

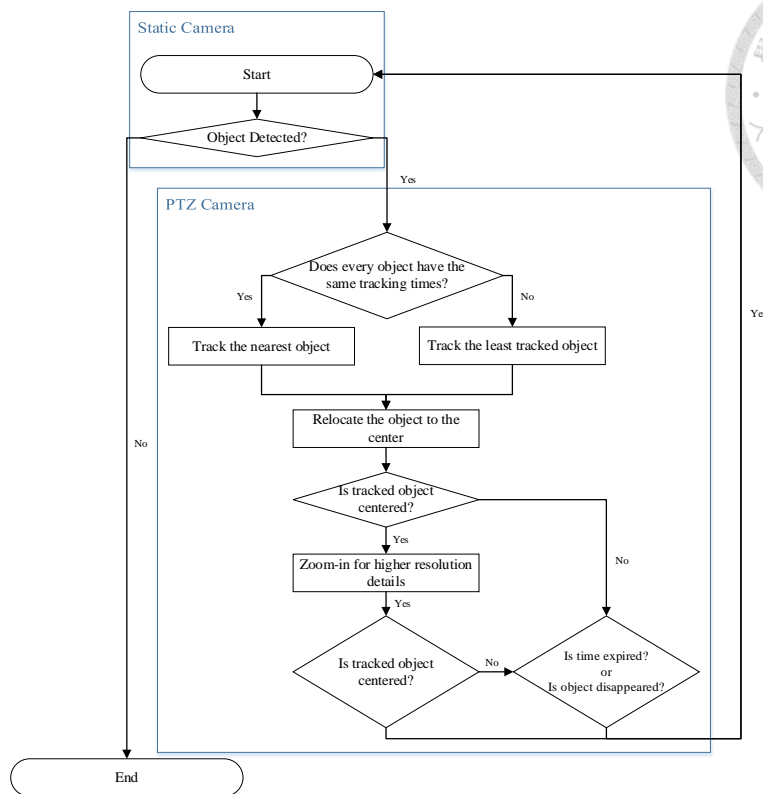
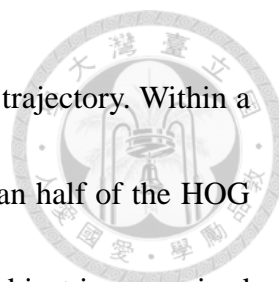


Figure 3-12 New PTZ tracking strategy of the system

Figure 3-12 is the new PTZ tracking strategy of our system. We make sure the tracked object is in the center of the image. To prevent an infinite loop occurs in the self-tracking algorithm, we break the algorithm loop when there is nothing to track on the screen or when the defined time of the timer expired. Although we lost the chance to get the other objects information, this new method helps us to get more usable one instead.

3.3.5 Human Detection and Human Face Detection

Since we aim to observe human activities, we check whether objects we detected are human or not before sending the locations to the PTZ camera. Every contour of the object is passed to the HOG classifier for human detection (Dalal and Triggs, 2005). We use a



simple voting method to decide if the object is a human for a single trajectory. Within a specific timeframe, a person will be detected as a human if more than half of the HOG features is recognized as a human being. Once the trajectory of the object is recognized as a human being, the predicted location will be sent to the PTZ camera.

Once the predicted location is sent to the PTZ camera, a HOG classifier is used to detect human face from the captured image of the PTZ after the relocating process from PTZ itself. If a face is successfully detected, the information will be added to the trajectory to associate same objects more precisely.

3.3.6 Simple Static Object Detection (SOD)

We use a simple method to track all the static objects and the environment changes, exclusive of the moving ones. To trace the environment changes, the fast self-tuning background subtraction algorithm is used. This algorithm is a background based object detection method. However, it returns every object that is not belong to the background. We then applied the Gaussian Mixture Model background subtraction (GMM) to solve the problem. GMM also returns the objects that is not belong to the background, but with a faster learning rate. In other words, the static objects will soon become the background of the GMM after a short period. We can define the period. Using the time difference of the two algorithms, we are able to separate the moving objects and the static objects while recording the environment changes.



Before subtracting the results of the two algorithm, some mechanics are applied to eliminate the noise and false alarms.

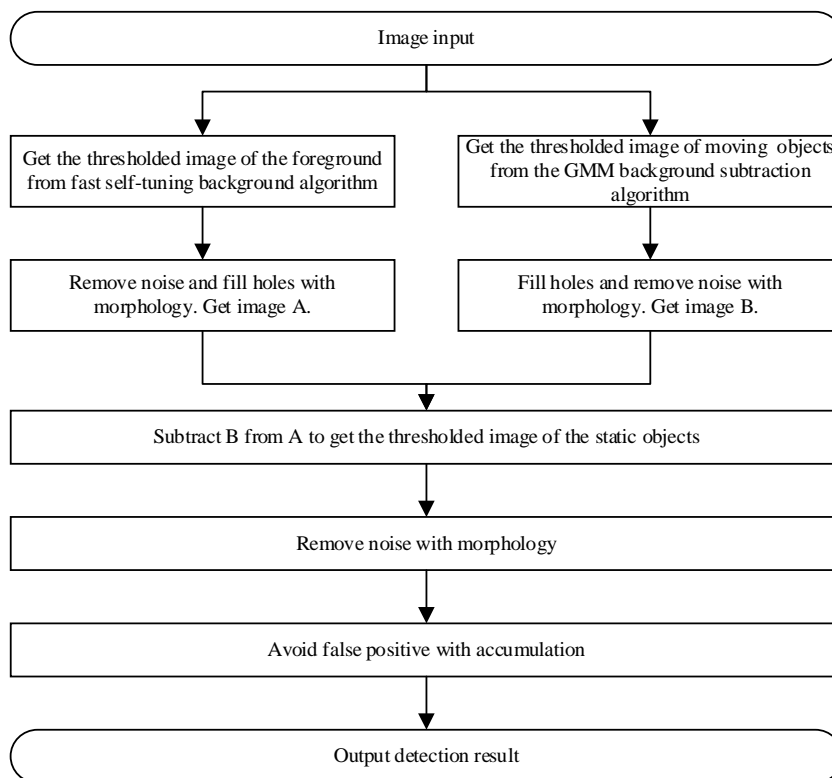
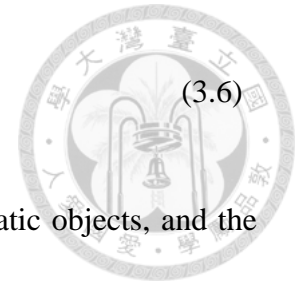


Figure 3-13 Diagram of Static Object Detection

Holes Filling and Noise Eliminations

First, we thresholded the background subtraction results. A simple erosion-and-dilation morphology method is applied to remove the noise (Figure 3-14). The holes from the fast self-tuning background subtraction algorithm are filled after removing the noise. Because the detection result from GMM contains more fragments, we dilate the GMM result with a kernel of size k to fill the holes before removing the noise. The size is determined according to the width and the height of the input frame.

$$k = \frac{\min(h, w)}{50} \quad (3.6)$$



The main purpose of the algorithm is to find the location of the static objects, and the correct position of an object is much more important than the contour of an object. After removing noise and filling holes of the results, we dilate the result of GMM a little bit to ensure the moving objects and the noise is completely remove from the final result.

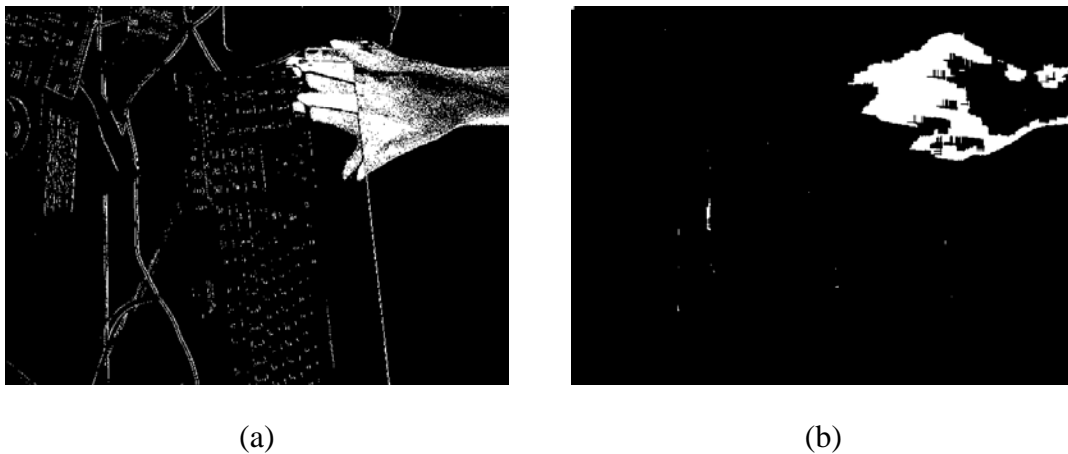


Figure 3-14 Noise Comparison between normal and morphology-fix. (a) Raw thresholded image from the GMM algorithm result. (b) Morphology-fixed image of (a).

Binary Image of the Static Objects

We subtract the GMM result (Figure 3-15 (a)) from fast self-tuning result (Figure 3-15 (b)) after the morphology step. If the pixel is smaller than the value zero, it will be set to zero. GMM results often produce more noise than the fast self-tuning background subtraction results and are dilated a little bit in the morphology step, so most of the noise will be removed after the subtraction of the two results. An erosion-dilation step will be

applied after the subtraction to reduce the noise again.

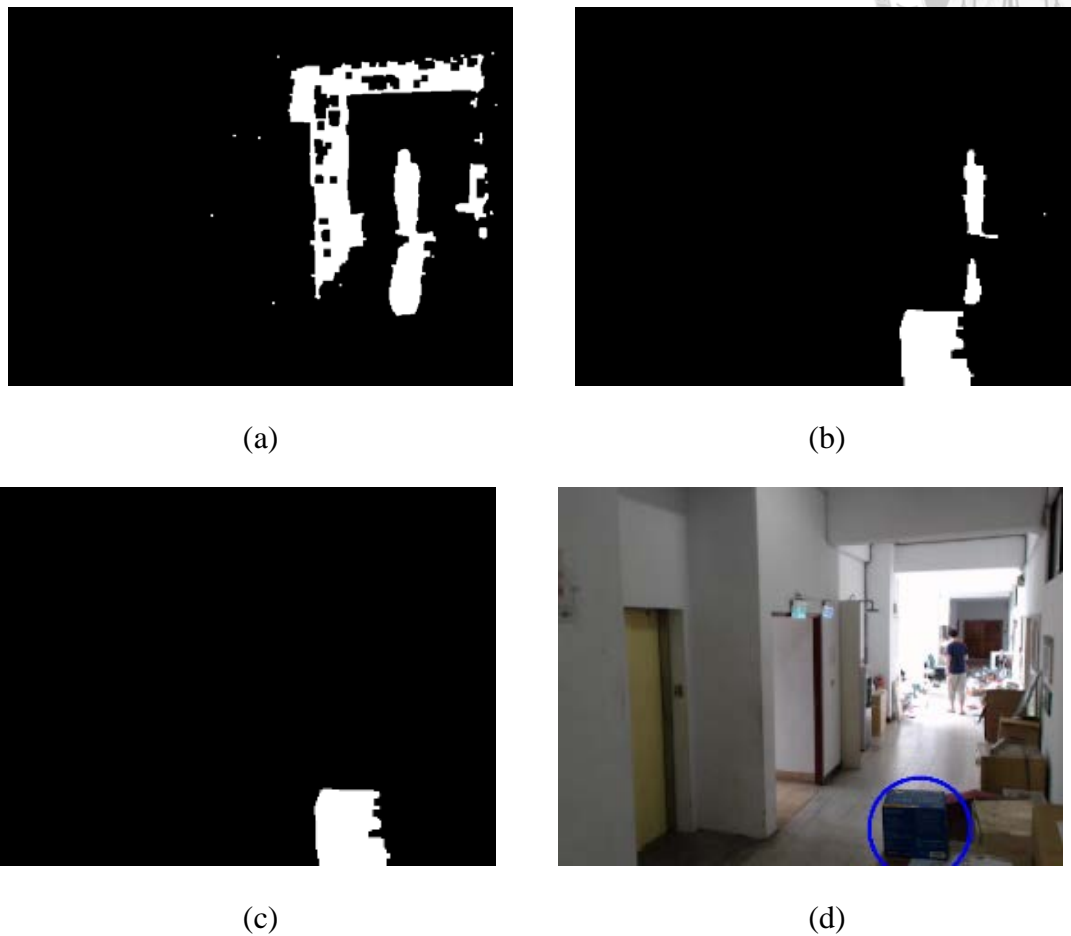


Figure 3-15 The process of the SOD algorithm. (a) GMM result. (b) fast self-tuning background subtraction algorithm result. (c) subtracting (b) from (a). (d) is the RGB version of static object detection result of (c).

Removing False Alarms with Accumulation

False alarms include a temporary static object and random severe noise. We use a simple accumulation map to make sure the static object stays long enough on the screen. Every pixel in the map is subtracted by 1 in every incoming frame, and is added by 2 if the incoming pixel is the center of the object. Because we use the contour finding to

calculate the center of a static object, considering the error produced, we also added the eight surrounded pixels by 2. The pixels of the accumulation map have a minimum value 0 and with no maximum limitation. The threshold ε is set to judge if the static object stays long enough according to the following equation.

$$\varepsilon = (\text{frame per second}) \cdot (\text{time in second}) \quad (3.7)$$

Initialization Time

The two algorithm used in this method have different initialization time. The GMM algorithm completes its initialization in nearly five frames. However, the fast self-tuning background subtraction algorithm requires more time to initialize. If there is no interference source while initializing, it will take about 60 frames to initialize. If the interference sources keep moving around in the foreground, the algorithm will take more time to build the background. Luckily, the algorithm can initialize in section. Any background of a section in the FOV will be built if it remains no interference for more than 60 seconds.

Initialization with Interference Sources

We have noticed that the fast self-tuning algorithm can initialize by section in order to deal with the interference in the real world. It is nearly impossible to find a place that has no interference at all. In the case with interference sources, as you can see in Figure 3-16 (a), there are two people walking on the road. The early-stage image (about 20

frames) in Figure 3-16 (b) shows that fast self-tuning background subtraction algorithm stills unable to initialize while the GMM algorithm has already completed initializing (Figure 3-16 (c)). A simple rule mentioned in chapter 2.5.2 shows that a pixel must have a background pixel in its neighborhood in order to become a background pixel itself. So the pixels remain stable from the edge to the center of the image will be built first, while the trajectory of the interference sources will remain unknown (Figure 3-16 (d)). Once the unknown section remains stable long enough, the pixel will be built as background.

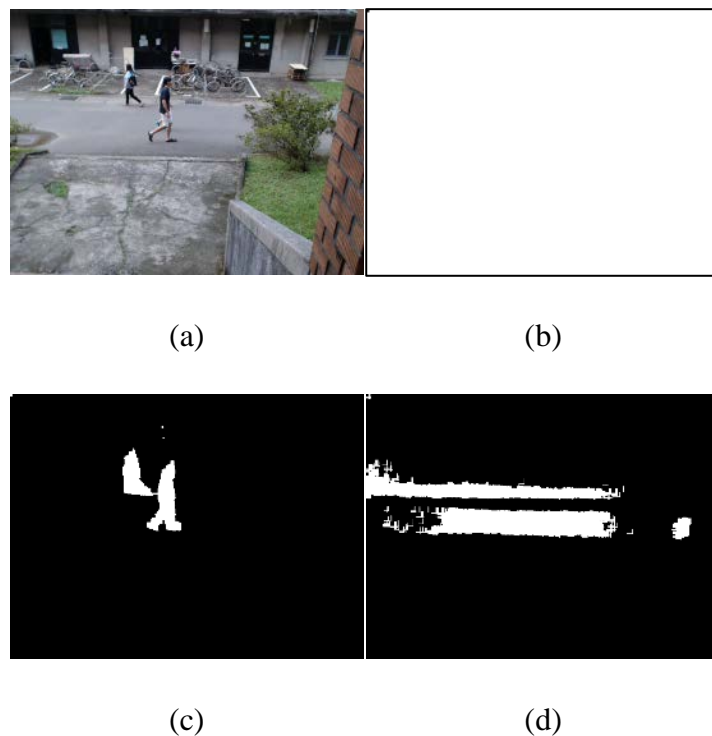
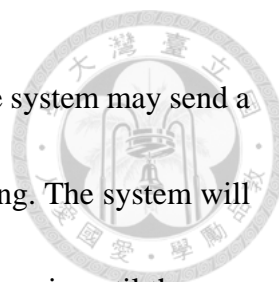


Figure 3-16 The image during the initialization stage of each algorithm. (a) Raw image. (b) Early-stage image of the fast self-tuning background subtraction algorithm. (c) Image of the GMM algorithm. (d) Initializing image (about 60 frames) of the fast self-tuning background subtraction algorithm.



If there is a static interference source in the initializing stage, the system may send a false alarm if the static interference source is gone after the initializing. The system will be able to self-correct the true background, but the false alarm will remain until the new background stays on the screen for more than a half compare to time that the interference source stays on the screen.



Figure 3-17 False alarm effect on fast self-tuning algorithm. (a) False alarm caused by a static interference source while using fast self-tuning background subtraction algorithm.

(b) A RGB version of (a).

3.3.7 Mixture of Static Object Detection and Object Tracking

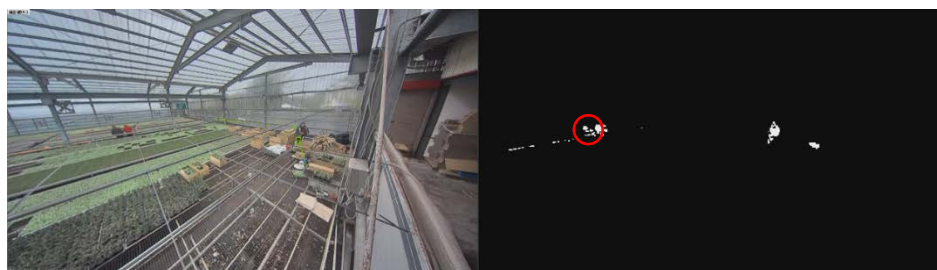
Since the Static Object Detection has the ability to detect still objects, we can solve the problem of losing tracking of object tracking method when a person stands still on the screen. The red circle at Figure 3-18 (b) shows that the GMM background subtraction fail to detect the foreground person wearing taupe jacket. The fragment detect result makes



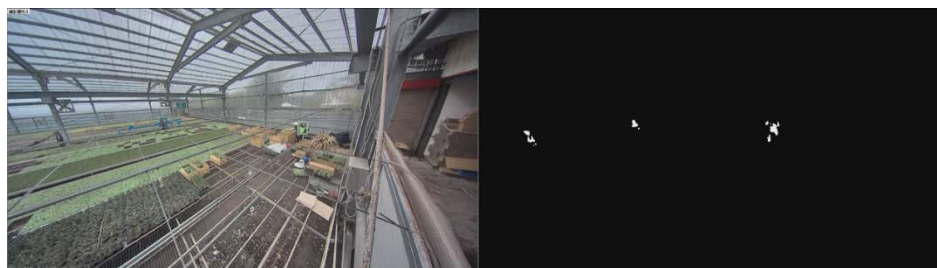
the system to treat it as noise. The system loses tracking of the man after the two people separated (Figure 3-18 (c)).



(a)



(b)



(c)

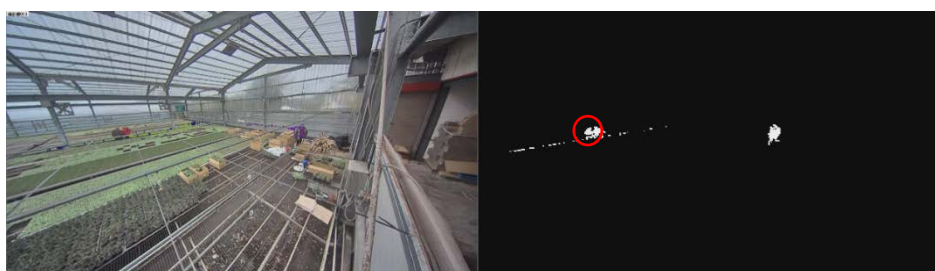
Figure 3-18 Object tracking process without SOD (a) the taupe and red jacket people at the top-left are detected by the system (b) the system fails to build the foreground because the taupe jacket man stops moving for a while (c) the system loses track of the taupe jacket man



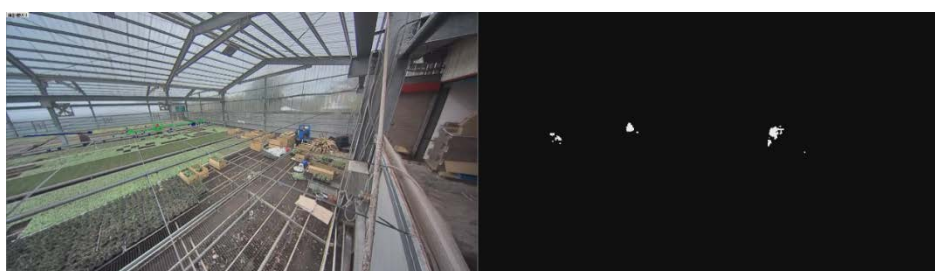
By checking whether the detected object is human or not of the Static Object Detection result, we can enhance the foreground area of the taupe jacket man as the red circle mentioned at Figure 3-19 (b). After the two people separated, the system is able to detect the taupe jacket man correctly as a tracked object at Figure 3-19 (c).



(a)



(b)



(c)

Figure 3-19 Object tracking process with SOD (a) the taupe and red jacket people at the top-left are detected by the system (b) the system successfully enhances the foreground of the taupe jacket man (c) the system is able to track both of the people after separate



3.3.8 Assignments of the Static Camera and PTZ

We are facing the issue of the architecture design of the system. The system is design to run in three threads in order to prevent different tasks blocking each other. However, the three threads are not synchronized that we are not able to share the data structures used different hardware. Currently the assignments of the static camera and the PTZ are as follows:

Static Camera:

1. Main object tracking
2. Trajectory details
3. Behavior recognition

PTZ:

1. Self-tracking to relocate the object to the center
2. Provide close-up identity information for each tracked object

3.3.9 Software Acceleration

In image processing, computing the images using GPUs is far faster than using CPUs. We choose to use the cross platform framework OpenCL to speed up our performance in heavy calculation steps. These steps include background based object detection and pre-processing of the incoming frames. Some steps like machine learning based algorithms will not be moved to the GPU because the time data transform from CPU to GPU might

longer than the total compute time. Although CUDA also uses GPUs as well, we dropped it because it only supports NVIDIA GPUs.



We also abandon to build panorama image and use ultra-wide FOV images instead.

Although panorama images have no blind spot, the process frame per second is very low that it is hard to observe the motion patterns of a place. We simply have our system installed at the corner of the room instead in the center to adapt the change of the static camera.



3.4 Data analysis

3.4.1 Choosing Working Patterns

Case 1 Zhi Chen farm in Taoyuan County

Our goal is to let the owners or the managers get to know their place better in order to improve and keep track of the working patterns. It is important to review the working patterns for the first day by hand. In Figure 3-20, we can summarize the patterns into four categories in the Zhi Chen farm, from the left to right includes:

1. **Plantlet operations** Plantlets operations including placing and taking the plantlets from the platform. Plantlets placed in this warehouse is ready for sale.
2. **Walking** Since the warehouse is built in the center of the Zhi Chen farm, one or more routes in the warehouse leads to other section of the farm. We can often see people passing here.
3. **Carrying plantlets** Workers here move the plantlets by putting them in a box. Carrying boxes around means they are ready or are carrying plantlets to other section in the farm.
4. **Talking, or other behaviors** Sometimes people stand in warehouse to look at the product for sale. Or they just stand there and do something else except plantlets operations.



Figure 3-20 The working patterns found in Zhi Chen farm. From the left to right are plantlets operations, walking, carrying plantlets, and others

Case 2 3rd Vegetable packaging factory in Taoyuan County

The working pattern of third vegetable packaging factory in Taoyuan County can be seen in Figure 3-21, from the left to right includes:

1. **Cargo Operations** Trucks and cargos staying in front of the factory entrance are ready to load the vegetables and take them away.
2. **Forklift Operations** Forklifts load the basket with vegetables and carry them around in the factory.
3. **Walking** People walking around in the factory during the open hours.
4. **Human Operations** People in front of the doorway of the factory are loading and unloading the vegetables on and off the forklifts and cargos.
5. **Passing in / out** The traffic flow of the vegetable packaging factory is very busy. It is important to record the trend of the place.



6. **Going home** The owner of the factory built his house inside of the factory field.

Using this system, we can get the activities of the owner and his family.

7. **Parking** The parking lot of the factory is located at the right in the cameras FOV.

We're able to see if there's any car parking there in a day.



Figure 3-21 The working patterns in the 3rd vegetable packaging factory in Taoyuan County

Case 3 Outside of the Tomatake Hall

We try to test our system by monitoring the traffic outside of the Tomatake Hall in National Taiwan University. The working pattern can be seen in Figure 3-21, from the left to right includes:

1. **Field activities** Besides monitoring the traffic, we try to see if there is any person wondering near the sand bunker.
2. **Staying under the overhangs of the building** We also try to figure out if there is any one staying under the overhangs of the building.



3. **Passing by- human** This is the main purpose of our test. We try to monitor the people passed by the road.
4. **Passing by- car** This is the main purpose of our test. We try to surveil the cars passed by the road.



Figure 3-22 The working patterns in the outside of the Tomatake Hall

Case 4 Plaza in front of the Dept. of Bio-industrial Mechatronics Engineering

We try to test our system by monitoring the traffic of the Plaza in front of the department of Bio-industrial Mechatronics Engineering (BIME) in National Taiwan University. The working pattern can be seen in, from the left to right includes:

1. **Go into the Building** We want to know if anyone goes into the Dept. of BIME
2. **Wandering** We would like to find out if someone is taking a nice walk or wandering around in the plaza.
3. **Passing by- human** This is the main purpose of our test. We try to monitor the people passed by the road.



4. **Passing by- car** This is the main purpose of our test. We try to surveil the cars passed by the road.



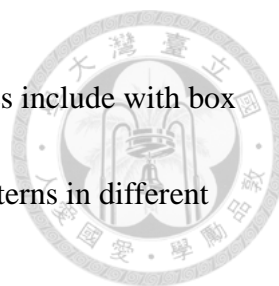
Figure 3-23 The working patterns in the Plaza of the Dept. of BIME

3.4.2 Finding Usable Features

Once the working patterns are decided, we have to find features to let the system able to classify the patterns. The raw information we can acquired from our system includes:

1. Happening time, duration of a tracked object
2. Trajectory points
3. Sub-image of a tracked objects for every frame
4. Captured faces acquired from PTZ

Time information and trajectory points information can be combined and produce trajectory features, and the sub-image information can produce behavior features. The face acquired by the PTZ is used to recognize the identity of a person. The trajectory



features include walking state and staying state. The behavior features include with box and no box. These four features are able to represent the working patterns in different combination shown in Table 3-5, Table 3-6.

Table 3-5 Working patterns represented by the combination of the features in Zhi Chen farm

	Trajectory Features	Behavior Features
Plantlets Operations	Staying	With Box
Carrying Plantlets	Walking	With Box
Walking	Walking	No Box
Others	Staying	No Box

Table 3-6 Working patterns represented by the combination of the features in 3rd

packaging factory



	Trajectory Features	Behavior Features
Cargo Operations	Staying in front of the factory doorway	Car
Forklift Operations	Staying	Forklift
Walking	Walking	Human
Operations at factory doorway	Staying in front of factory doorway	Human
Passing in / out	Walking	Transportations (car, motor)
Going home	Staying at home spot Walking to home spot	Human
Parking	Staying at parking lot	Car, motor

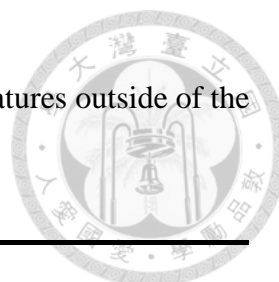


Table 3-7 Working patterns represented by the combination of the features outside of the

Tomatake Hall

	Trajectory Features	Behavior Features
Passing by- car	Walking on the road	Car
Passing by- human	Walking on the road	Human
Field activities	Walking at field	Human
Staying under the overhangs of the building	Staying under the overhangs	Human

Table 3-8 Working patterns represented by the combination of the features in the Plaza

in front of the Dept. of BIME

	Trajectory Features	Behavior Features
Passing by- car	Walking	Car
Passing by- human	Walking	Human
Go into the building	Walking near stairs	Human
Wandering	Staying somewhere in the plaza	Human
Parking	Stay	Cat



Trajectory Features

Since we have no tags on our tracked objects for us to calibrate and locate them precisely, the trajectories acquired from the static camera are full of noise. The paths are too jittered to use them directly just like the path in Figure 3-24.

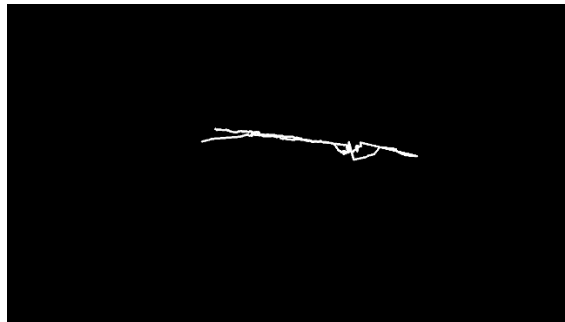


Figure 3-24 A severely jittered path

We solved this by giving pre-defined map points to simplify the raw trajectories. In Figure 3-25, you can see seven points pointed on the roads of the warehouse. With the time and the duration recorded with every trajectory points, we can calculate whether a person stays at a spot (one of the seven chosen points) for a long time. The threshold for staying action is determined by the following equation:

$$\frac{\text{stop by time of any same points}}{\text{total time of all the trajectory points}} > \text{threshold} \quad (3.8)$$



Figure 3-25 The pre-defined map points in Zhi Chen farm. These points help us remove the noise of the trajectories

The noise and the jittering of a path is removed by using these pre-defined points to represents the path Most of the too detailed trajectories points will be removed and only the trend of the path will be left. We now use a sequence of numbers to represent the path instead of using a list of x, y coordinate. The system also provide custom define motions which is used to make these points into groups. Some keywords are used for system to recognize the meanings of the points on the map (Figure 3-31 left), such as “entrance”, and “crossRoad”. Note that the map points here will be resize if the resolution is downgraded. In this case, we downgrade from 1080p to 720p in order to speed up the process of the static camera algorithm.

	X	Y	label name	define motion
1	471	292	NA	NA
2	563	306	NA	NA
3	428	324	NA	NA
4	270	351	NA	NA
5	38	391	NA	NA
6	678	330	NA	NA
7	863	361	NA	NA



Figure 3-26 The setting panel of the pre-defined map points. “label name” and “define motion” are editable cells.

Behavior Features

We use Gabor filter and Bag of Words to process every sub-image (Figure 3-27) of a tracked object for every trajectory point. Each of the processed image has their own name named with the name of its trajectory. We use the supervised machine learning methods to train and predict those images, including K-Nearest Neighbor, Naïve Bayes, SVM, and Neural Network. In the case of the Zhi Chen farm, the label for every image is categorized into “WithBox” and “NoBox”. At last, we use voting methods with the predicted images to determine the results for each trajectory to increase accuracy.



Figure 3-27 An example of a tracked sub-image, labeled “WithBox”



3.4.3 Mixture Behavior Features (Gabor and Bag of Words)

Bag of Words has better accuracy in three or more classifications. Its endurance of the data variant is also larger than Gabor filter. Unfortunately, Bag of Words uses SIFT feature points as its descriptor. Images that are too small cannot acquire SIFT features (Figure 3-28). We tried different sizes of images to test the accuracy of the Bag of Words and are shown in Figure 3-29.

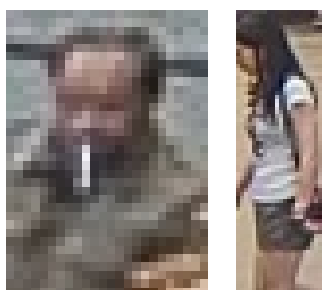


Figure 3-28 Acquired sub-images with different resolutions. Although we are able to recognize the object in the left image is a person, SIFT fails to produce features. The right image has a higher resolution and is able to produce SIFT features.

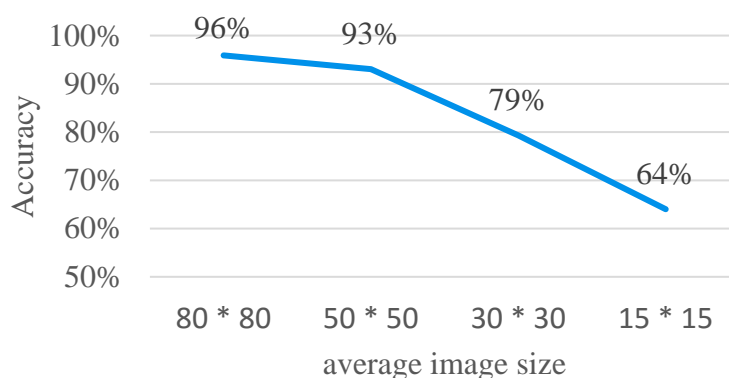
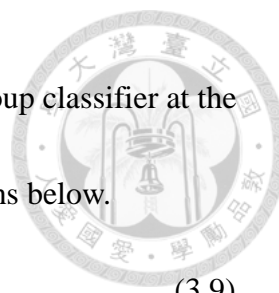


Figure 3-29 Different image size effect on Bag of Words



Since Gabor filter is able to recognize the object and use a 2-group classifier at the accuracy of 80%, we use multiple algorithms following the definitions below.

$$\begin{cases} \text{if image size} > 50 \cdot 50, \text{ use Bag of Words} \\ \text{else, use Gabor filter} \end{cases} \quad (3.9)$$

3.4.4 Feature Classifications

We use the open-source analysis tool Orange to build a network to train those images. Orange is a fast prototype analysis tool by simple drag-and-drop coding to test the possibility of an idea. The trained classifier can be saved for further use. The node “Test Learners” lists all the machine learning methods with their accuracy and other information for users. We are able to check if our features are chosen correctly and which method is the most suitable for us. Once the features are tested, we can use them in our system to predict behaviors.

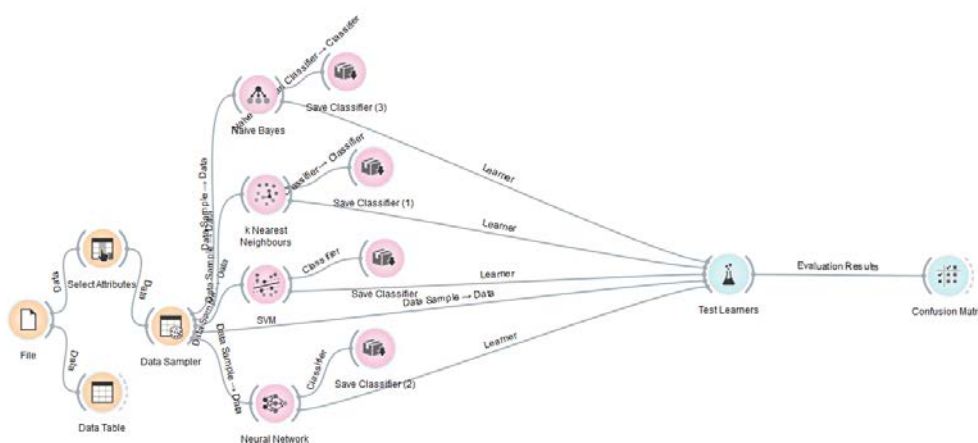


Figure 3-30 Fast prototyping and training using Orange



3.4.5 Custom Define Rules System

In order to make the system expandable and able to fit most of the environment, we support using custom define rules system to create new working patterns in modules without modifying the code. The new working patterns can be created with any combination of the raw information, pre-defined map points, and existing working patterns. You can even use a user-defined name as rules for more advanced analysis.

These new defined rules are saved in the format of comma-separated CSV files.

	X	Y	label name	define motion
1	473	291	entranceA	A
2	566	309	crossRoad	A
3	438	330	toCross	NA
4	237	363	toEnC	C
5	16	397	entranceC	C
6	728	342	CB	B
7	938	378	entranceB	B

Read Rules

Save Rules

+ -

	rule name	type	rule
1	Plantlet Operations	TYPEwBEHAVIOR ▾	Stay,carryingbox
2	straight	POSITIONDEFINE ▾	A,B,B,A
3	turn	POSITIONDEFINE ▾	A,C,C,A;B,C;C,B

Figure 3-31 The setting panel of the custom define rules

With this feature, we are able to do a quick setup even in our own experiments. Here we will explain the setup in detail for each experiment sites. Currently, we provided nine types of define rules to let users to create new custom rules with the existing rules:

1. STARTTIME

This option provides users to create rules depend on the start time of the detected objects.



2. ELAPSETIME

This option provides users to use the duration information in the data structure to create custom rules.

3. POSITION

The custom rules are depended by the name of the map points only.

4. POSITIONDEFINE

The custom rules are depended by the group of the map points only.

5. TYPE

The custom rules are depended by the type of the trajectories only.

6. BEHAVIOR

The custom rules are depended by the sub-image classification results only.

7. TYPEwBEHAVIOR

This mixture define provides users to create custom rules with both trajectories and sub-image classification results.

8. ELAPSETIMEwPOSITION

This mixture define provides users to create custom rules with both duration and the name of the define map points.

9. TYPEPOSITIONwBEHAVIOR

This mixture define provides users to create custom rules with trajectories, name of

the map points, and the sub-image classification results.

10. TYPEPOSITIONDEFINEwBEHAVIOR

11. This mixture define provides users to create custom rules with trajectories, group of the map points, and the sub-image classification results.



Zhi Chen farm in Taoyuan County

Table 3-9 is the setup of the pre-defined map points in the Zhi Chen farm. With the map points and the data structure information such as trained sub-image features, we are able to create our custom rules to meet our needs. Group of the pre-defined map points categorize points into road. Same group names means that they belong to the same road. The custom rules for Zhi Chen farm are shown in Table 3-10.

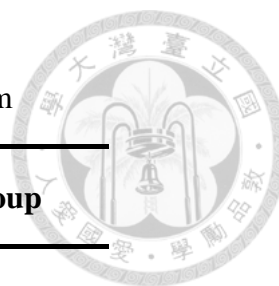


Table 3-9 Pre-defined map points of Zhi Chen farm

Map Points	Coordinate	Point Name	Group
1	(473, 291)	entranceA	A
2	(566, 309)	crossRoad	A
3	(438, 330)	toCross	(N/A)
4	(237, 363)	toEnC	C
5	(16, 397)	entranceC	C
6	(728, 342)	CB	B
7	(938, 378)	entranceB	B

Table 3-10 Custom rules for Zhi Chen farm

Custom Rule	Type	Define Rules
Plantlets Operations	TYPEwBEHAVIOR	Stay carryingbox
Carry	TYPEwBEHAVIOR	Walk carryingbox
Walk	TYPEwBEHAVIOR	Walk nobox
Others	TYPEwBEHAVIOR	Stay nobox
Go straight	POSITIONDEFINE	A B;B A
Turn	POSITIONDEFINE	A C;C A;B C;C B

3rd Vegetable packaging factory in Taoyuan County

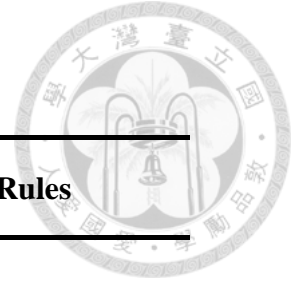


Because the experiment of the packaging factory is a large field, we did not group any point. We only focus on the “most stay position” analysis and the daily and weekly summaries.

Table 3-11 Pre-defined map points of the Packaging Factory

Map Points	Coordinate	Point Name	Group
1	(457, 339)	entranceA	(N/A)
2	(337, 371)	entranceB	(N/A)
3	(181, 450)	entranceC	(N/A)
4	(585, 360)	4	(N/A)
5	(559, 432)	5	(N/A)
6	(492, 557)	6	(N/A)
7	(803, 365)	home	(N/A)
8	(806, 435))	8	(N/A)
9	(814, 565)	9	(N/A)
10	(966, 370)	parking	(N/A)
11	(1070, 432)	entranceD	(N/A)

Table 3-12 Custom rules for Packaging Factory



Custom Rule	Type	Define Rules
Cargo Operations	TYPEwBEHAVIOR	Stay car
Forklift	TYPEwBEHAVIOR	Walk toragon
Walking	TYPEwBEHAVIOR	Walk human
Staying Operation	TYPEPOSITIONwBEHAVIOR	Stay human entrancA
Passing Car	TYPEwBEHAVIOR	Walk car
Passing Motor	TYPEwBEHAVIOR	Walk motor
Passing Motor 2	TYPEwBEHAVIOR	Walk motor2
Go home 1	TYPEPOSITIONwBEHAVIOR	Stay human home
Go home 2	TYPEPOSITIONwBEHAVIOR	Stay human home

Outside of the Tomatake Hall

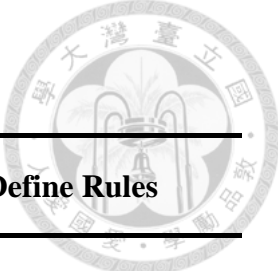


This experiment is to test the traffic monitoring ability of our system. We use group names to analyze the pass-by cars and humans. We also try to figure out the around usage of the surroundings.

Table 3-13 Pre-defined map points of the Tomatake Hall

Map Points	Coordinate	Point Name	Group
1	(265, 471)	RoadMostLeft	A
2	(456, 434)	RoadLeft	A
3	(684, 383)	RoadMid	A
4	(934, 334)	RoadRight	A
5	(1104, 260)	RoadMostRight	A
6	(427, 326)	BuildingFront	B
7	(677, 281)	Playground	B
8	(722, 592)	TomatakeEntrance	T

Table 3-14 Custom rules for Tomatake Hall



Custom Rule	Type	Define Rules
Passing by- human	TYPEPOSITIONDEFINEwBEHAVIOR	Walk A human
Passing by-car	TYPEPOSITIONDEFINEwBEHAVIOR	Walk A car
Field Activities	TYPEPOSITIONwBEHAVIOR	Walk playground human
Stay under the overhangs of the building	TYPEPOSITIONwBEHAVIOR	Stay BuildingFront human

Plaza in front of the Dept. of BIME

This experiment is to test the traffic monitoring ability of our system. We use group names to analyze the pass-by cars and humans. In this case, we use pre-defined maps to remove noises. Point 1 to 15, 30, 38 are noise points.

Table 3-15 Pre-defined map points of the plaza in front of the Dept. of BIME

Map Points	Coordinate	Point Name	Group
1	(109, 137)	noise	noise
2	(321, 105)	noise	noise

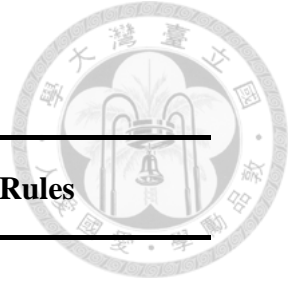
(Continued)

(Continued)



15	(182, 222)	noise	noise
16	(934, 334)	parkingFront	road
17	(1104, 260)	parkingLot	road
18	(427, 326)	stairway	road
19	(677, 281)	stairFront	road
20	(722, 592)	roadEntranceA	road
21	(524, 400)	A-B3	road
22	(632, 385)	A-B2	road
23	(651, 336)	roadEntranceB	road
24	(699, 353)	B-C1	road
25	(767, 360)	B-C2	road
26	(857, 376)	B-C3	road
27	(977, 384)	B-C4	road
28	(1092, 397)	roadEntranceC	road
29	(1196, 402)	roadEntranceC-2	road
30	(1093, 534)	noise	noise
38	(655, 675)	noise	noise

Table 3-16 Custom rules for the Dept. of BIME



Custom Rule	Type	Define Rules
Passing by- human	TYPEwBEHAVIOR	Walk human
Passing by-car	TYPEwBEHAVIOR	Walk car
Wandering	TYPEwBEHAVIOR	Stay human
Go into the Building	TYPEPOSITIONwBEHAVIOR	Walk human stairway; Walk, human, stairFront
parking	TYPEPOSITIONwBEHAVIOR	Stay cat parkingFront; Stay car parkingLot

3.4.6 Working Pattern Analysis and Daily and Weekly Summary

The simplified trajectories and behavior features will be used to predict action of every trajectory. A daily summary in the format of CSV file is produced every day with 24 hours a day recording the appearance, status, custom rules, and comments. Weekly summary records the difference between days and the total sum of all the daily summaries.

3.4.7 Object View Manager

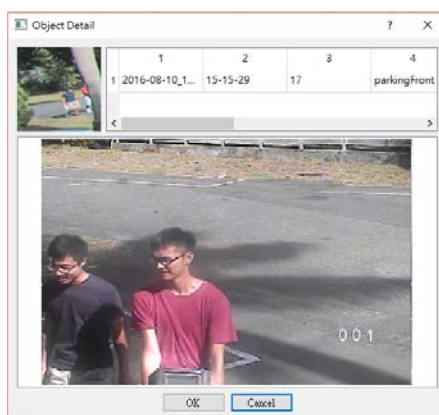
We provide a friendly GUI (Figure 3-32a) for users to look at the individual working analysis results just like the other commercial software. The Object View Manager has an internal clock to regularly search for the analysis results from the Cooperative



Surveillance system. Each of the object can be double-clicked to see captured PTZ images (Figure 3-32 b).



(a)



(b)

Figure 3-32 GUI of the Object View Manager (a) is the object list (b) is the object detail

3.5 Experiment

3.5.1 Experiment Sites

The system is set in the following places to test the performance.



Table 3-17 Experiment sites and the purposes

Experiment sites	Purpose
Greenhouse of National Taiwan University	object tracking testing
Inside and outside of the Tomatake Hall	Static Object Detection testing PTZ Self-tracking Algorithm
Zhi Chen farm in Taoyuan County	system performance testing
3rd Vegetable packaging factory in Taoyuan County	system performance testing
Plaza in front of the Dept. of BIME	System performance testing

Greenhouse of National Taiwan University



We make use of the greenhouse in the National Taiwan University to do the simple test of our system functions.



Figure 3-33 The panorama view of the NTU greenhouse

As we can see in Figure 3-33, the NTU greenhouse is simple, and with less interference. It is a perfect place to do testing experiments. Figure 3-34 shows the view of the static camera installed in the system.



Figure 3-34 The view of the static camera in the system



Inside and Outside of the Tomatake Hall

We choose an indoor case and an outdoor case of the Tomatake Hall to test the Static Object Detection algorithm. The static object detection is used to remove moving foreground and noise. Only the static objects are left. Figure 3-35 (a) and (b) show the indoor and outdoor case of the Tomatake Hall.



(a) Indoor case

(b) Outdoor case

Figure 3-35 Two cases of the Tomatake Hall

Another spot we choose is the end of the corridor in the second floor of the Tomatake Hall. We setup the system there and record the people walking on the street. Figure 3-36 (a) is the road that appears in the record video. Figure 3-36 (b) and (c) is the different view of the setup system.



(a)



(b)



(c)

Figure 3-36 The scene and the setup of the PTZ self-tracking algorithm

Plaza in front of the Dept. of BIME

The plaza in front of the Dept. of BIME (Figure 3-37) is surely a nice place to test our system, the view here is clear enough to gather the information we need. Unfortunately, this is an outdoor experiment, which will be severely effect by shadows in sunny days and water reflections in rainy days.



Zhi Chen farm in Taoyuan County

Zhi Chen farm is the first place to test if the whole system works properly and useful.

We choose this warehouse to observe how and when the farmers here works to sale the plantlets. We fortunately have a very well placed position for the system to record the activities (Figure 3-38 (a) and (b)). Figure 3-38 (c) and (d) show the view of the two different sections of the farm split by the view of our system.



(a) Front view of the system



(b) Back view of the system



(c) The warehouse taken from section A



(d) The main road of section A in the

to B

warehouse

Figure 3-38 The experimental site of Zhi Chen farm



3rd Vegetable packaging factory in Taoyuan County

This is the second place we choose to test our system. We try to analyze the working pattern of the traffic control in front of the packaging factory. We aim at to figure out when and how do the workers here take the packaged vegetable for sale. Figure 3-39 (a) shows the front view of the system, and Figure 3-39 (b) shows the doorway of the factory, where most of the vegetable loading operations are done here.



(a) The front view of our system (b) The doorway of the packaging factory

Figure 3-39 The experimental site of the 3rd vegetable packaging factory

3.5.2 Object Tracking

We tested the improved object tracking method in NTU greenhouse. It is an ideal place with the least interference for us to test our system. Two cases were tested include horizontal and vertical movement on the screen in the farm. The horizontal movement tests the constancy of the object tracking and the vertical movement test the effect of the dynamic threshold for the Euclidean distance.

3.5.3 Static Object Detection

To prove the algorithm works, some simple experiment is done by choosing an indoor and outdoor position with total 6 cases: three indoors, and three outdoors. Each position is verified one object detection, multiple object detection with interference sources, and multiple interference sources only.



Chapter 4 Results and Discussion



4.1 System Performance

4.1.1 Simple Object Tracking

The scene and the trajectory of a tracked human in the greenhouse of National Taiwan University in vertical and horizontal directions are shown in Figure 4-1. Due to the noises in the GMM subtraction model, the trajectory of a human is sometimes jittered.

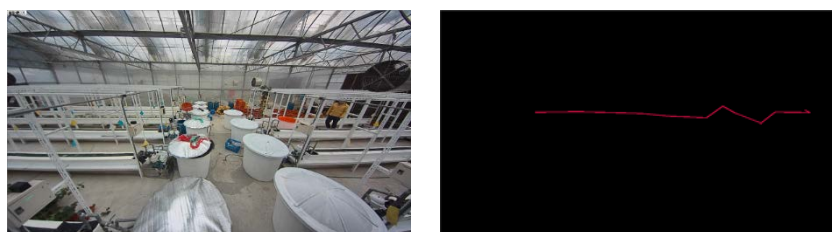


Figure 4-1 Simple object tracking demonstration. The trajectory (right) drawn by the system when a person's movement is detected (left).

We use two simple methods to test our performance of the simple object tracking. First, the multi-resolution GMM method is used to reduce the noise of the foreground detections. We test the performance by the following equation,

$$\text{noise ratio} = \frac{\text{noise object}}{\text{total detections}} \quad (4.1)$$

We pick randomly 30 frames and get a result of 8%. As the GMM without multi-resolution, we get a result of 21%.



Second, the purpose of the dynamic Euclidean threshold is used to prevent the improper disconnection due to the object distance to the lens. Thus, this method is checked by the following equation,

$$\text{disconnect rate} = \frac{\text{disconnected trajectories}}{\text{total trajectories}} \quad (4.2)$$

The result is shown in Table 4-1. The reason of the improvement is simple; the fixed Euclidean distance must be set to a fixed value that may not work in other distance. Fixed Euclidean distance gets an average minimum error but is not the best performance. Be cautions that neither of these two methods are able to adapt great speed changes of the moving objects.

Table 4-1 The disconnect rate comparison between fixed and dynamic Euclidean threshold

Distance to the camera lens	20m	15m	10m	5m
Fixed threshold	5%	4%	13%	20%
Dynamic threshold	6%	7%	9%	8%



4.1.2 PTZ Image Mapping to PTZ Sphere Coordinate

When the point is selected in the PTZ image, it can be mapped to the sphere coordinate. After that, the PTZ is moved to the position with the estimated pan the tilt angles. We can see if the selected point is located in the center of the image. With the following equation, we can calculate the performance of our system,

$$error = \frac{\sqrt{((x_{estimated} - \frac{w}{2})^2 + (y_{estimated} - \frac{h}{2})^2)}}{\sqrt{(\frac{w}{2})^2 + (\frac{h}{2})^2}} \quad (4.3)$$

where the $(x_{estimated}, y_{estimated})$ is the new position of the selected point in the previous pan and tilt angles.

The first test is to test the performance of the value Z (from equation 3.4). Because the tilt value will affect the result, we use zero as the tilt angle to test the entire zooming rate to minimize the tilt effect. We randomly pick 5 point for each zooming rate. From Figure 4-2, we can see that the errors for each zooming rate is smaller than 2%.

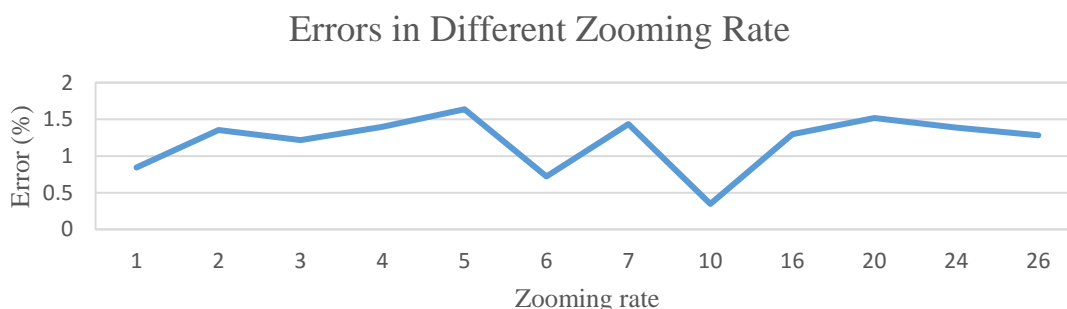


Figure 4-2 Errors in Different Zooming Rate

The second test is to find out the effect from the tilt angle. From Figure 4-3, we can see



that when the tilt angle is zero, the error rate of distance to the center of the image is hardly affected by the tilt angle. When the tilt angle comes to 80, the error rate is severely affected by the tilt angle. The error rate and distance to the tilt angle relationship results comes from the average error rate of 5 points for each distance. This is reasonable because the geometry transform equations between x-y plane image and the PTZ sphere contain arc tangent. The variable put in the arc tangent are the tilt angle and the distance on the image. The error rate will be affected by the tilt angle and the distance.



Figure 4-3 Errors caused by the tangent effect.

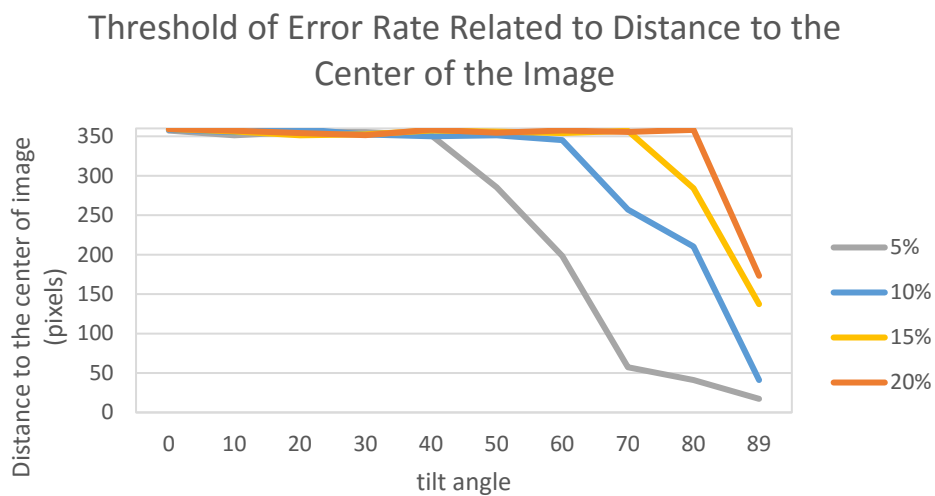


Figure 4-4 Threshold representation of the tilt angle for each error rate. The x-axis is the the tilt angle and the y-axis is the distance to the center of the image, which the maximum distance to the center of image is about 360 pixels.

Figure 4-4 shows the angle threshold of the system. If the error rate smaller than 5 % is desired, we have to set an angle threshold equals to 40 for the system to ensure the mapping function works as us expect. For any detected object that is larger than the threshold, we have to consider moving the PTZ twice or more to locate the object in the center. However, if we choose to accept the error rate smaller than 20%, then for any tilt angle smaller than 80 degrees are acceptable for us. Figure 4-5 shows the error rate of the angle threshold of the PTZ when the distance to the center of the image is 350 pixels.

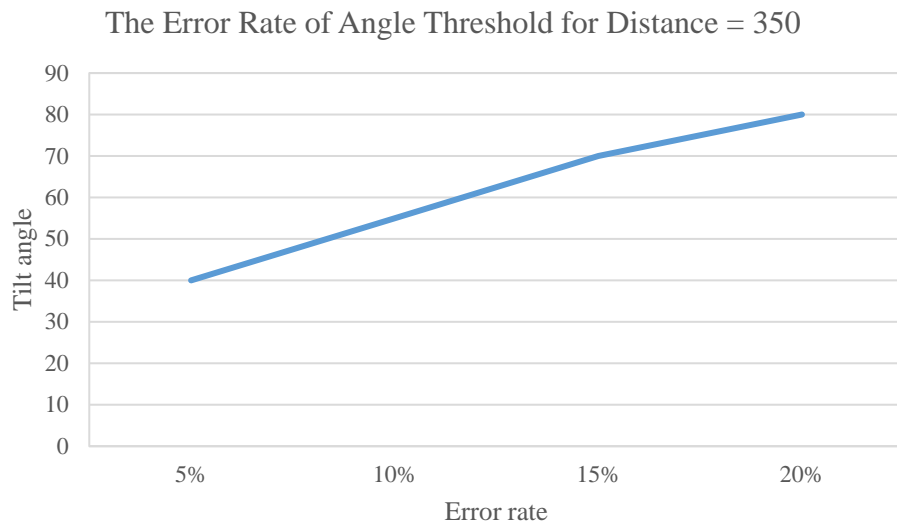
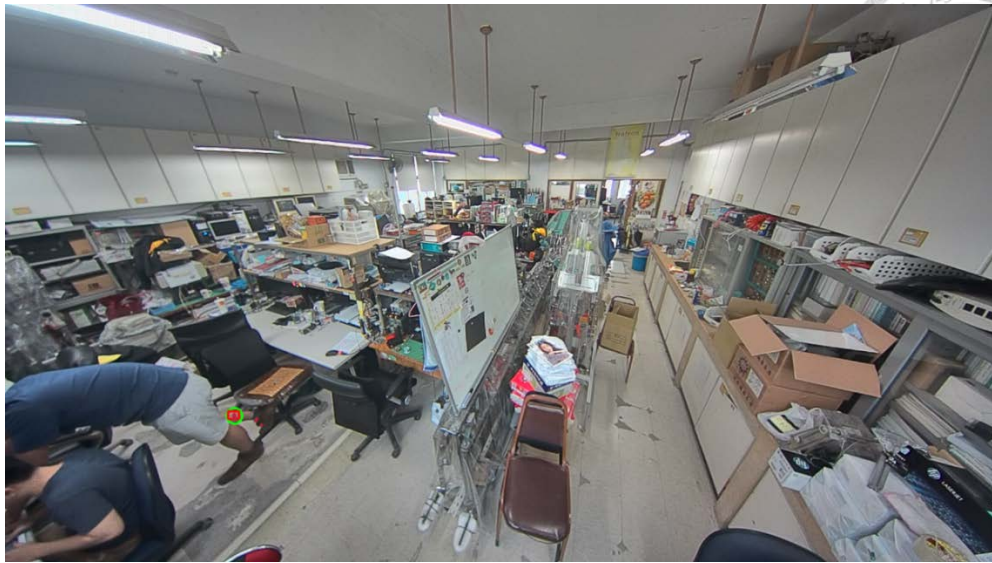


Figure 4-5 The error rate of angle threshold for distance about 350 pixels of different tilt angle

4.1.3 PTZ Self-tracking Algorithm

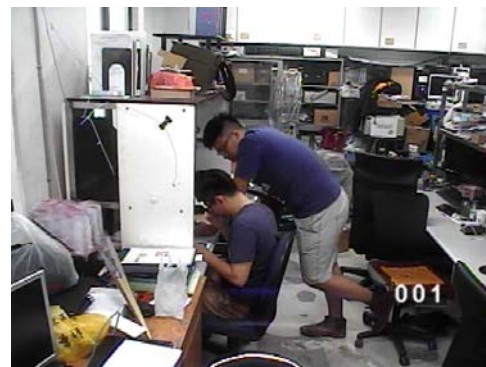
Figure 4-6 shows a very great example of how the PTZ self-tracking mechanism works. There are two reasons tell us that why this function is important. First, sometimes static camera fails to track the whole object due to the effect of noise or part of the tracked object is not moving. In Figure 4-6 (a) tells that only the man's feet is moving in the view of the static camera. Second is that the geometry transform between cameras has error that fails to locate the PTZ into the right position. Figure 4-6 (b) shows the result when PTZ receive the estimated coordinate from the static camera.



(a)



(b)



(c)

Figure 4-6 Example of PTZ Self-tracking algorithm (a) is the view of the static camera (b) is the estimated pan-tilt view from the static camera of the PTZ (c) is the correct pan-tilt view

Due to these two reasons, the result does not satisfied our needs. However, in Figure 4-6 (c) shows that the PTZ redirect the view to the right position when it found something is moving in the pan-tilt view. The performance of the self-tracking algorithm can be reviewed by two index, the mapping performance and the centroid estimation. The



mapping performance is told in the last section. The performance of the centroid estimation is about 40 pixels (30 trials, image size 720x480) calculated by the following equation

$$error = \sqrt{(X_{centroid} - X_{real})^2 + (Y_{centroid} - Y_{real})^2} \quad (4.4)$$

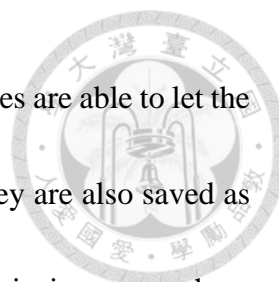
where X_{real}, Y_{real} are the real tracked object coordinate define by ourselves, $X_{centroid}, Y_{centroid}$ are the computed coordinate.

4.1.4 Acquired Images from PTZ

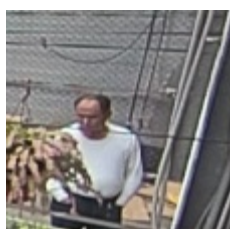


Figure 4-7 Image captured from PTZ

Figure 4-7 show some example of close-up look from the PTZ camera. The information provides us able to check out the details from the Object View Manager when needed. The PTZ has some advantage and some drawbacks. We use the PTZ images in Zhi Chen farm (Figure 4-8) to explain the advantage more detailed. The PTZ is able to track and zoom in to get the detail of the tracked object. Figure 4-8 (c) shows the track



human and Figure 4-8 (d) shows the human carrying box. These images are able to let the managers and owners to identify the tracked object more clearly. They are also saved as usable features in the database of our system, especially the zoom-in images such as Figure 4-8 (d). Figure 4-8 (d) makes up the insufficient information of Figure 4-8 (b).



(a)



(b)



(c)



(d)

Figure 4-8 The tracked object in static camera and its PTZ image in Zhi Chen farm

Unfortunately, due to errors of the background subtraction and geometry transform between cameras, sometimes the predicted coordinate does not locate the tracked object at the center of the PTZ image. Figure 4-9 (a) shows a tracked human walking in the warehouse of the Zhi Chen farm. The estimated coordinate is sent to the PTZ. The PTZ will return a result of Figure 4-9 (b), which the detected object is not centered in the PTZ

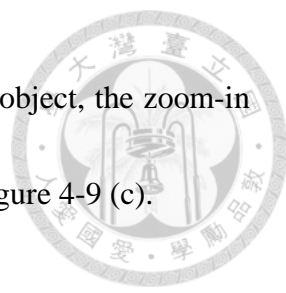


image. If the self-tracking algorithm is not able to self-centered the object, the zoom-in image will still fail to get the detail of the tracked object shown in Figure 4-9 (c).



(a)



(b)



(c)

Figure 4-9 Fail case when zoom in object is not at the center of the PTZ image

Hardware Limitations

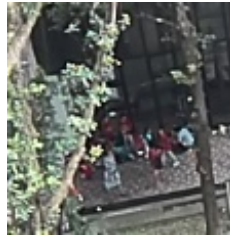
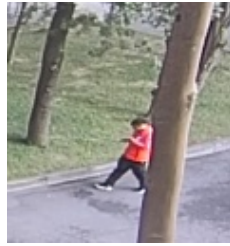


Figure 4-10 PTZ self-centered the detected crowded objects and zoom in to the center of the image

Although we develop a PTZ tracking strategy to capture the object for further details, the limitations of hardware still restrict us to get every information we needed. The motor of the PTZ installed in our system does not move very fast and cannot change its destination during moving. This cause a problem that if the object moves too fast related to the zooming rate of the PTZ or if the object moves in unpredictable ways, the PTZ is very likely to lose site of the object. Figure 4-10 is an example of nearly static crowded objects that only moves in a certain area randomly. The PTZ is able to easily track and center them.



(a)

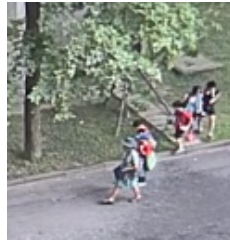


(b)

Figure 4-11 Tracing single object with PTZ

However, Figure 4-11 and Figure 4-12 shows the limitations of the hardware. Figure 4-11 (b) shows that when the PTZ tries to zoom in to look at the man looking at his cellphone, the man can easily walk out of the sight of the PTZ. With our tracking strategy, we have to start all over again to request a new estimated coordinate from the static camera to relocate the man.

Fortunately, if we tried to track multiple objects in the same time Figure 4-12 (a), it is harder to lose sight of them because the zoom in rate will be small enough to prevent this situation happens. Figure 4-12 (b) shows that PTZ is able to zoom-in and self-centered the crowds without losing them.



(a)



(b)

Figure 4-12 Tracing multiple objects with PTZ

We do a little test to find out the limitation of our PTZ installed in our system. The total reaction time of PTZ for every command is about 1 second. Which means the object cannot move exceed $720/2=360$ pixels in x-axis or $480/2=240$ pixels in y-axis in one second. We then can use the information from the IMM filter and Kalman filter to get the estimated velocity and direction to find the suitable zooming ratio for the PTZ system and can be concluded into the following equation,



$$\begin{aligned}
 (x_{new}, y_{new}) &= \vec{V}_{es} * t + (x, y) \\
 (\theta_{new}, \phi_{new}) &= G(x_{new}, y_{new}), (\theta, \phi) = G(x, y) \\
 (d\theta, d\phi) &= (\theta_{new} - \theta, \phi_{new} - \phi) \\
 Z_1 &= \frac{\frac{w}{2}}{\tan^{-1}(d\theta)}, Z_2 = \frac{\frac{h}{2}}{\cos^{-1}(d\theta) \tan^{-1}(d\phi)} \\
 \text{zooming rate} &= \frac{\min(Z_1, Z_2) * \text{CCD width}}{\text{focal} * w}
 \end{aligned}
 \tag{4.5}$$

where w, h are the width and the height in the unit of pixels of the PTZ image, \vec{V}_{es} is the estimated velocity vector in the unit of pixels from the static camera, t is the reaction time of the PTZ camera, Z is the parameter introduced in equation 2.2, $focal$ is the focal length of the PTZ, and CCD width is the CCD of PTZ in the unit of mini-meters.

4.1.5 Human Face Detection by PTZ

The PTZ camera zoomed-in to the estimated location to collect the information of any detected person and captured the face (Figure 4-13). The face is clear and recognizable in the accuracy of 40% (20 test faces). The misdetections are caused by the size of the tracked human and the viewing angle of the face. Faces that are too small or tilt is not recognizable.



Figure 4-13 The acquired faces from the PTZ image



The system still has drawback although it is able to capture the faces of the tracked human. Cause the limitation of the PTZ camera, we are not able to trace the object and change the view of the PTZ at the same time. With the errors from the estimated location, sometimes it is quite easy to lose track of the object when the object is not centered while the PTZ makes a close up look.

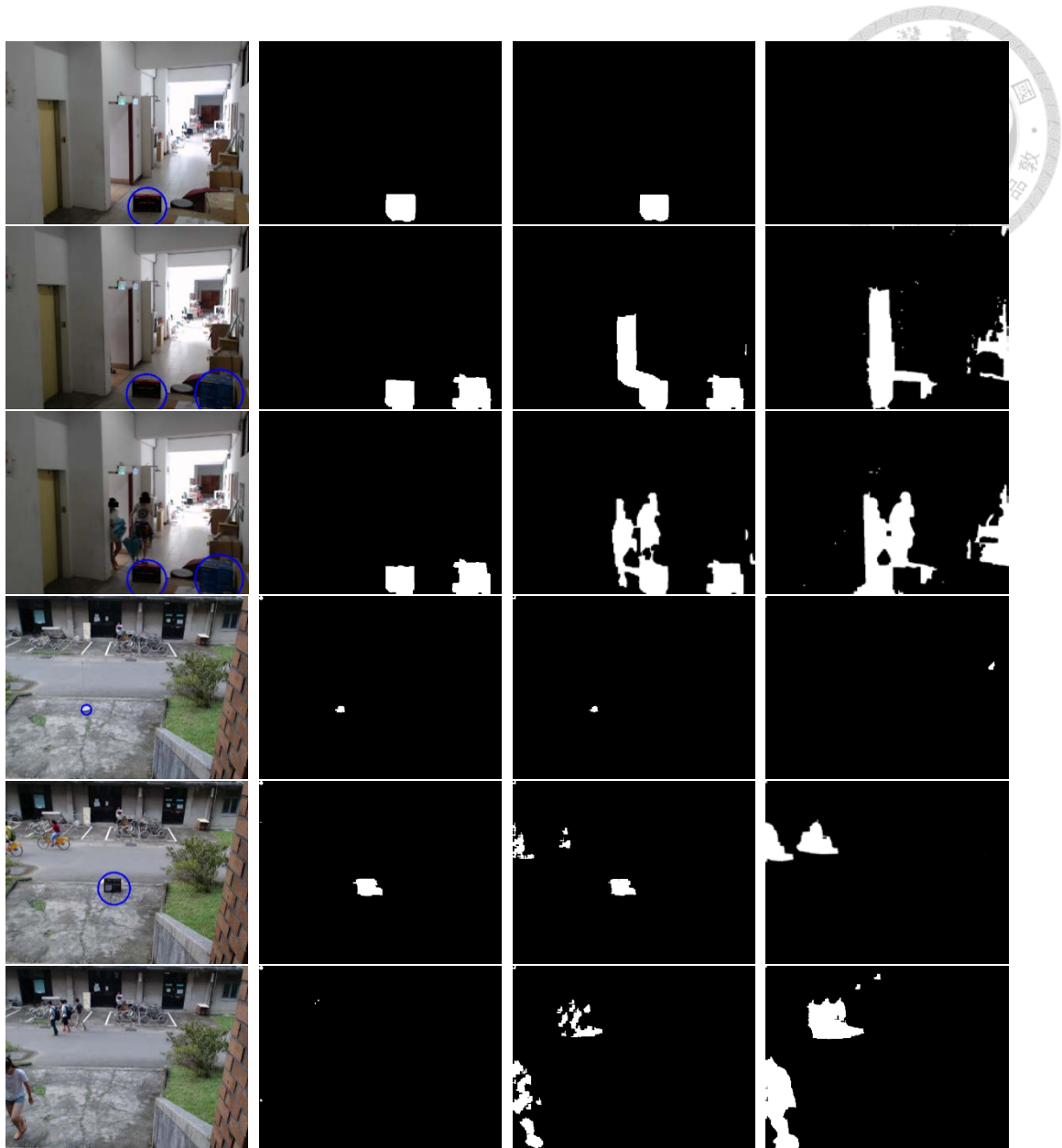
4.1.6 Static Object Detection

The main purpose of this method is to find abandon objects or scene modifications.

The following table is the results. Any interference while initializing is exclusive.

Table 4-2 The results of static object detection

Place	Cases	Success / Total	Success rate
Total		30 / 40	75 %
Indoor	General Cases	10 / 10	100 %
	With Static Interference	5 / 10	50 %
Outdoor	General Cases	9 / 10	90 %
	With Static Interference	6 / 10	60 %



(a) RGB results (b) Subtract (d) from (c) (c) Fast Self-tuning (d) GMM Result

Background

Subtraction Result

Figure 4-14 The results of the static object detection. The 6 cases in sequence are: indoor single static object detection, indoor multiple static object detection, indoor multiple static object detection with multiple interference sources, outdoor single static object detection, outdoor multiple static object detection with multiple interference sources, and multiple interference sources only



Case 1 Indoor Single Static Object Detection

This is the simplest and realism case. Because indoor lights are far more stable than outdoors, so the object is easily detected.

Case 2 Indoor Multiple Static Object Detection

The objects are all successfully detected. They all passed the accumulation test.

Case 3 Indoor Multiple Static Object Detection with Interference Sources

This case has people walking around. The GMM result successfully caught the moving object only and the fast self-tuning background subtraction caught the foreground. The static objects were left after the subtraction of the two results.

Case 4 Outdoor Single Static Object Detection

This is also a simple case, except the noise from the light source may affect the detection result. But the fast self-tuning background subtraction algorithm is able to against a certain degree of noise, we're able to successfully caught the object.

Case 5 Outdoor Multiple Static Object Detection with Interference Sources

The reason is same as case 3, except for the noise from the light, which is same as case 4.

Case 6 Outdoor Interference Sources Only

This case is to verify that the system will not send a result if there is no static object.



False Alarm

Because the method we used will catch any static object on the screen, including a standstill person (Figure 4-15). This will cause a false positive. However, this can be fixed by combining this method with other human detection algorithms such as the HOG classifier we used in our system.



Figure 4-15 False alarm demonstration of the SOD algorithm. (a) The result of the detection. (b) The RGB version of (a)

4.1.7 Mixture of the Static Object Detection and Object Tracking

We use additional algorithms to prevent trajectory disconnected. Static Object Detection continues the trajectory if the tracked object is temporarily static on the screen. Table 4-3 shows the disconnect rate of the algorithm. We can see that the disconnect rate has slightly decrease.

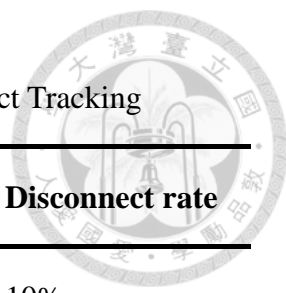


Table 4-3 The performance of the Mixture of SOD and Object Tracking

	Disconnect trajectories	Total trajectories	Disconnect rate
With SOD	5	50	10%
Without SOD	7	50	14%

4.2 Data Analysis

4.2.1 Choosing Machine Learning Method

Zhi Chen Farm

We try to figure out if the detected object has a box with him or not (Table 3-5). We find a series of images of human with box and another group of images of human with no box (Figure 4-16), and train them with four different supervised machine learning methods.

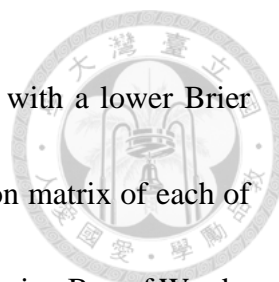


(a) human with box

(b) human with no box

Figure 4-16 Features chosen to classify between “with box” and ”no box”

The 2-feature training result using Gabor filter of the success rate is shown in Table 4-4. The four machine learning methods have almost the same accuracies. Although Naïve Bayes has the highest accuracy, the Brier value is also the highest. We then decided



to use SVM instead because its accuracy is the second highest but with a lower Brier value. Before making the final decision, we will look at the confusion matrix of each of the learning method, and the accuracies of the feature training results using Bag of Words.

Table 4-4 Gabor filter. 2-Feature training results of Zhi Chen farm. Cross validation (number of folds 10). Using first day images as our training data (74).

	Accuracy	Brier
kNN	78.21%	0.3044
Naïve Bayes	82.68%	0.3464
SVM	79.82%	0.3167
Neural Network	79.82%	0.3343

Table 4-5 is the confusion matrix of four machine learning method. Although kNN has a well accuracy, it has a low accuracy on detecting people with “no box”. The other results of the machine learning methods look just as same as the accuracy shown in Table 4-4, at the accuracy about 80%.



Table 4-5 Gabor filter. Confusion matrix of the feature training results from Zhi Chen farm.

kNN			SVM		
	With Box	No Box		With Box	No Box
With Box	89.2%	18.9%	With Box	81.1%	18.9%
No Box	32.4%	67.6%	No Box	21.6%	78.4%

Naïve Bayes			Neural Network		
	With Box	No Box		With Box	No Box
With Box	78.4%	21.6%	With Box	81.1%	18.9%
No Box	13.5%	86.5%	No Box	21.6%	78.4%

Table 4-6 shows the accuracies of the 2-feature classifier using Bag of Words. There are two possibilities that makes the overall accuracies higher than the results using Gabor filter. First, the training image size of Bag of Words is larger than Gabor filters which contains more details. Second, Bag of Words uses SIFT feature points which is more robust to recognize the same object in different viewing direction. In this case, SVM has the highest accuracy; the confusion matrix of SVM shown in Table 4-7 also has great results with high accuracies.

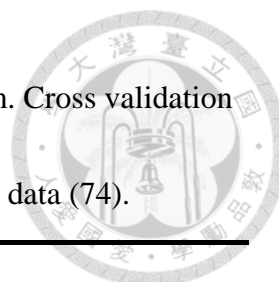


Table 4-6 Bag of Words. 2-Feature training results of Zhi Chen farm. Cross validation (number of folds 10). Using first day images as our training data (74).

	Accuracy	Brier
kNN	82.14%	0.2818
Naïve Bayes	87.50%	0.2385
SVM	95.89%	0.0818
Neural Network	93.39%	0.1304

Table 4-7 Bag of Words. Confusion matrix of the feature training results.

kNN	With Box	No Box	SVM	With Box	No Box
With Box	97.6%	2.4%	With Box	98.3%	1.7%
No Box	36.1%	63.9%	No Box	8.3%	91.7%

Naïve Bayes	With Box	No Box	Neural Network	With Box	No Box
With Box	89.2%	10.8%	With Box	98.3%	1.7%
No Box	13.9%	86.1%	No Box	13.9%	86.1%

We then choose SVM as our classifiers to train our dataset. Among the four machine learning methods and two different features classifier results, SVM has the highest



accuracy in average. The confusion matrixes of SVM is also close to its own accuracy.

Now we will verify our decision with the data in the next case.

3rd vegetable packaging factory

Because we are able to acquire bigger images in the packaging factory, the behaviors in the factory is also enough for us to train 5-feature classifiers with Bag of Words, here the Gabor classifier is a 2-feature classifier, and Bag of Words is a 5-feature classifier.

The accuracy of the 5-feature Gabor feature classifier is seriously bad, so we abandon it.

Table 4-8 Gabor filter. 2-Feature training results from 3rd vegetable packaging factory.

Cross validation (number of folds 10). Using first day images as our training data (210).

	Accuracy	Brier
kNN	80.48%	0.2796
Naïve Bayes	82.38%	0.3512
SVM	82.38%	0.2974
Neural Network	73.81%	0.4235

In Table 4-8, we can see the accuracy results of using Gabor features. The accuracy of SVM is also about 80 percent just like the accuracy in the previous place. The confusion matrix in Table 4-9 gives a good explanation of the Brier values in Table 4-8.

Naïve Bayes has almost the same accuracy as SVM, but it fails to detect if the object is a



human or not. The confusion matrixes of SVM and kNN also explain the phenomenon.

While SVM has a higher accuracy compare to kNN, but the variance between “Car” and

“Human” in SVM is larger than the values in kNN.

Table 4-9 Gabor filter. Confusion matrix of the feature training results from 3rd

vegetable packaging factory.

kNN	Car	Human	SVM	Car	Human
Car	86.6%	13.4%	Car	76.5%	23.7%
Human	24.8%	75.2%	Human	12.4%	87.6%

Naïve Bayes	Car	Human	Neural Network	Car	Human
Car	92.8%	7.2%	Car	72.2%	27.8%
Human	26.5%	73.5%	Human	24.8%	75.2%

Table 4-10 shows the accuracy results of a 6-feature classifier using Bag of Words.

The enough details from the vegetable packaging factory make us able to get an accuracy

of 80 percent in SVM. The accuracies of other machine learning methods are so low that

we will skip their confusion matrixes later. These accuracies may due to the following

reasons. kNN predicts the results by the determined k-value and the distribution of the

data among the space. If the variance of the input data is fuzziness, or if a wrong k-value

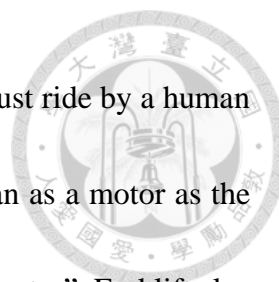


is determined, kNN will fail to make predictions. Naïve Bayes decides the group by using Bayes theorem. It is quite data dependent that it will be better to do a 2-feature classification in this case while the feature of the trajectories is too complicated. At last, Neural Network requires large amount of data to train. Our method to train the classifier using the first day data maybe is not enough for Neural Network for making a 6-feature classifier.

Table 4-10 Bag of Words. 6-Feature training results from 3rd vegetable packaging factory. Cross validation (number of folds 10). Using first day images as our training data (297).

	Accuracy	Brier
kNN	43.34%	0.7275
Naïve Bayes	66.32%	0.6430
SVM	81.16%	0.2946
Neural Network	75.63%	0.3582

The confusion matrix of SVM in Table 4-10 can be shown in Table 4-11. Each positive percentage looks well. The best result is detecting “Car” with an 89.1 percent accuracy, and 9.1 percent is detected as dog is because the training data of dogs and cars used in the packaging factory is black. The lowest accuracy comes from detecting



“Human”. This may cause by the training data. Since every motor must ride by a human and can be seen by the camera, it is very easy to misjudge the human as a motor as the SIFT features of the human is also a part of the features in the label “motor”. Forklift also has the same problem because the driver seat and the driver can also be seen from the camera videos.

Table 4-11 Bag of Words. Confusion matrix of the feature training results of SVM from 3rd vegetable packaging factory. Motor 2 is motor with platform.

SVM	Car	Dog	Human	Forklift	Motor	Motor2
Car	89.1%	0	1.8%	0	1.8%	7.3%
Dog	0	80.0%	20.0%	0	0	0
Human	0	1.4%	74.0%	2.7%	19.2%	2.7%
Forklift	0	1.6%	9.8%	77.2%	1.6%	9.8%
Motor	0	0	8.5%	0	87.3%	4.2%
Motor 2	0	0	4.5%	4.5%	13.7%	77.3%

Outside of the Tomatake Hall

The setup position of the hardware is a bit failure cause that the acquired sub-images are quite small. These small images only can be processed by the Gabor filter. Although from Table 4-12 we can know that SVM is not the best choice for this case, the accuracy



of the SVM is similar to other cases. This means that SVM is stable enough for most cases and will not provide randomly low-high accuracies in different cases.

Table 4-12 Gabor filter. 2-Feature training results from outside of the Tomatake Hall (number of folds 10). Using first day images as our training data (55).

	Accuracy	Brier
kNN	81.67%	0.2552
Naïve Bayes	75.00%	0.4990
SVM	86.00%	0.3353
Neural Network	81.67%	0.3121

From Table 4-13 we can conclude the problem of the low accuracy using SVM in this case. The Gabor filter seems to misclassify the car to human in 25.9%.

Table 4-13 Gabor filter. Confusion matrix of the feature training result of SVM from

outside of the Tomatake Hall

SVM	Car	Human
Car	74.1%	25.9%
Human	21.4%	78.6%



Plaza in front of the Dept. of BIME

From Table 4-14 we can know that SVM is not the best choice for this case. But as the same reason as the one in the Tomatake Hall, SVM has similar results to other cases, which is stable enough to use. Table 4-15 shows the confusion matrix of the SVM result.

Table 4-14 Gabor filter. 2-Feature training results from plaza in front of the Dept. of BIME (number of folds 10). Using morning (to 12:00PM) first day images as our training data (210).

	Accuracy	Brier
kNN	89.00%	0.1849
Naïve Bayes	68.33%	0.6333
SVM	77.33%	0.2870
Neural Network	75.67%	0.3316

Table 4-15 Gabor filter. Confusion matrix of the feature training result of SVM from plaza in front of the Dept. of BIME

SVM	Car	Human
Car	84.8%	15.2%
Human	25.0%	75.0%




In the above four cases, we are able to choose SVM as our classifier since the accuracy of the result is at 80 percent and because the most stable one in all machine learning methods. Since the viewing angle and visibility caused by weather is not always fine and is not able to control, the accuracy of 80 percent is acceptable and is able to increase the overall accuracy by using multiple features from the same trajectories with voting method to make the result better.

Table 4-16 The overview of all the accuracies

	Case 1		Case 2		Case 3	Case 4
	Gabor	BOW	Gabor	BOW	Gabor	Gabor
ACC	(2)	(2)	(2)	(6)	(2)	(2)
kNN	78.21%	82.14%	80.48%	43.34%	81.67%	89.00%
Naïve Bayes	82.68%	87.50%	82.38%	66.32%	75.00%	68.33%
SVM	79.82%	95.89%	82.38%	81.16%	76.00%	77.33%
Neutral Network	79.82%	93.39%	73.81%	75.63%	81.67%	75.67%

4.2.2 Overall Accuracies and Misdetections



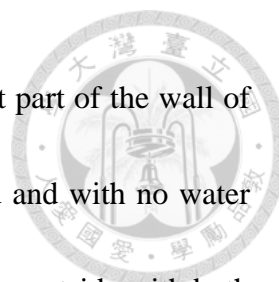
Although the training results of the SVM classifier has the accuracy at 80 percent, the reality of the overall accuracy of the other days (exclude the first day training data) is not always stable. When creating the classifier, we opt to choose the features that is able to recognize by ourselves. Which means we will not label a noise or a fuzziness feature and train them since we cannot determine the label for the feature. When analysis the other days' data, we feed the system with every sub-image acquired from each frame for each trajectories. These sub-images may include misdetections such as noise, disconnected trajectories, or wrong tracking objects. Voting method is able to limited remove these misdetections if those happened seldom. Unfortunately, voting method is not able to remove high frequency noise or constant interference cause by the weather condition. The accuracy and misdetection can be seen in Table 4-17 and Table 4-18. Case 4 does not have weekly overall accuracy because it is a one-day experiment. It has an accuracy of 74 percent.



Table 4-17 Overall accuracy of the vegetable packaging factory using raw trajectories

		sub-images						
		Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
	Origin	77%	74%	76%	72%	74%	74%	70%
Case 1								
	Voting Method	86%	81%	81%	78%	81%	80%	77%
	Origin	75%	67%	71%	72%	69%	68%	70%
Case 2								
	Voting Method	78%	73%	76%	77%	75%	74%	75%
	Origin	76%	71%	70%	72%	71%	65%	72%
Case 3								
	Voting Method	77%	74%	74%	76%	72%	69%	75%

The nature weather is a big problem to any image processing system. The static camera we have in our system has the function of HDR and white balance. We can roughly ignore the effect cause by the natural light. However, shadows in sunny days and water reflections in rainy days (Figure 4-17) is not removable. Water reflection is a high frequency and random noise and is hardly to remove from our system. Shadows of a tracked object is a constant interference and is a challenge to remove from our foreground.



In Zhi Chen farm, the roof of the warehouse is none transparent, but part of the wall of the warehouse is transparent. The effect from the shadow is limited and with no water reflection noise. In the vegetable packaging factory, the scene is taken outside with both noise severely haunted.



Figure 4-17 Water reflection (left) and shadow hazard (right)

From Table 4-18 we can see that rainy day and sunny day actually cause lots of noise and trajectories, which directly decrease our accuracy. The effect of the noise can be removed by the voting method. However, it is hard to increase the overall accuracy if the trajectories disconnect into pieces. Disconnected trajectories are hard to make voting decisions because the voters are limited.

Table 4-18 Misdetections cause by noise and disconnected trajectories (2016/04/29

(Case 2))

	Numbers of detection	Noise	Disconnected trajectories
Rainy day	551	63	134
Sunny day	663	48	107

4.2.3 Working Pattern Analysis

Our final goal is to visualize our data and present them the owners and the managers.

Our system produces results shown in Figure 4-18. According to the recording time set by the user, the system gives the information of the numbers of people for each working patterns in the unit of hours. The bar chart gives the user the result of the long-time observation of a place and is able to tell the user the trend of a place.



Zhi Chen farm in Taoyuan County

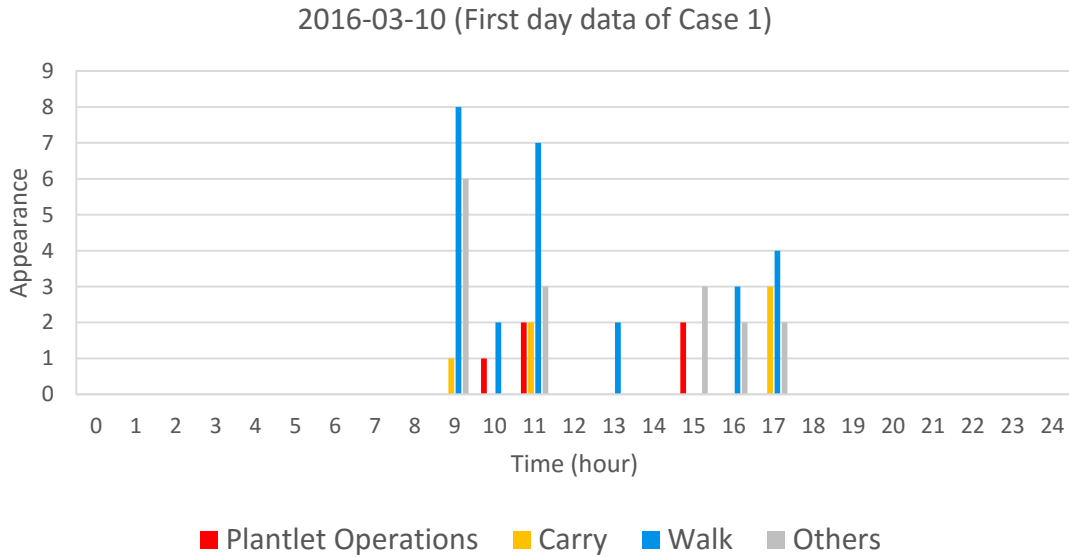


Figure 4-18 The daily result from Zhi Chen farm of the first day data

Figure 4-18 is the first day data of the Zhi Chen farm. The y-axis is the appearance of human and the x-axis is the time of 24 hours of a day. The system is record from 9 AM to 5 PM. We can see that most of the farmers works in the morning and in the afternoon and rest at noon. Furthermore, most of the operations are done in the morning.

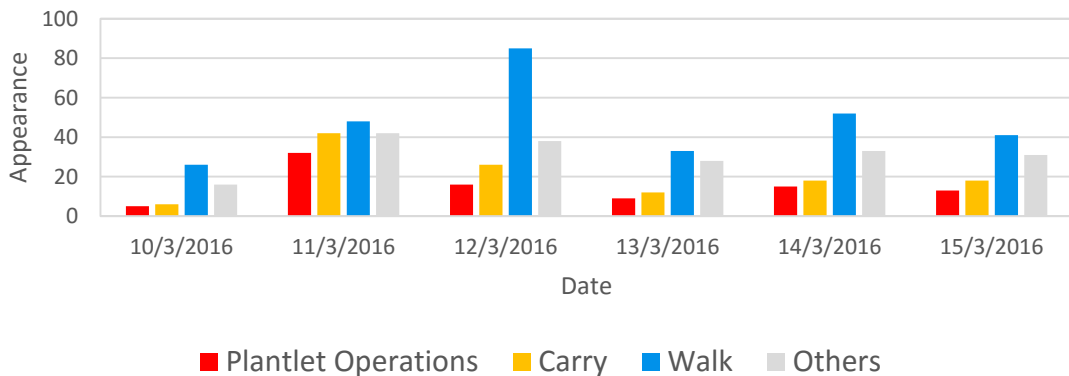
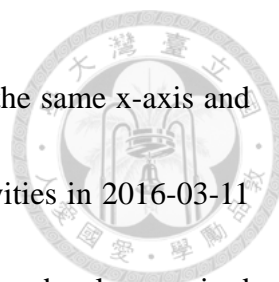
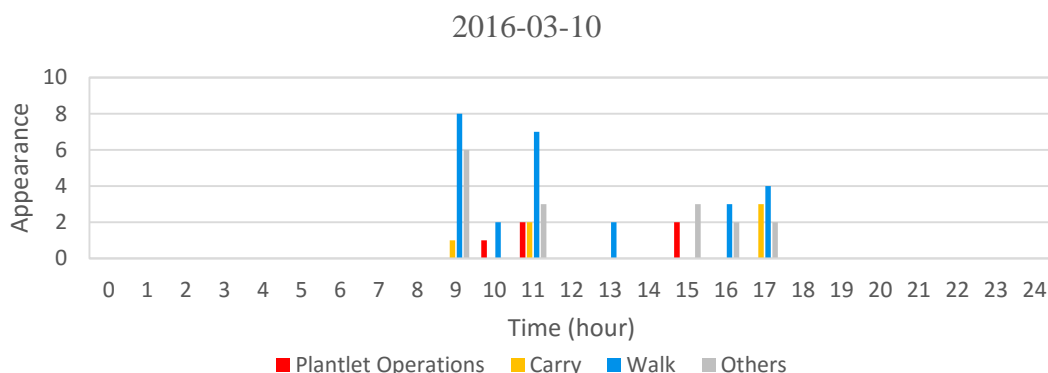


Figure 4-19 Weekly summary of Zhi Chen farm

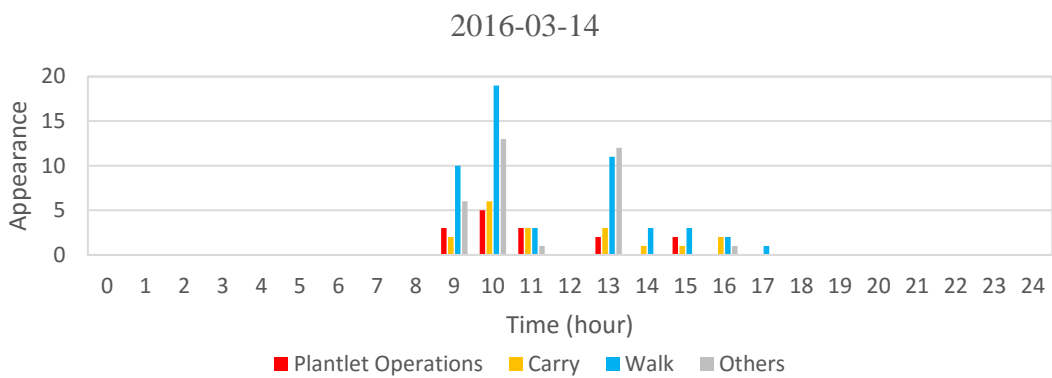
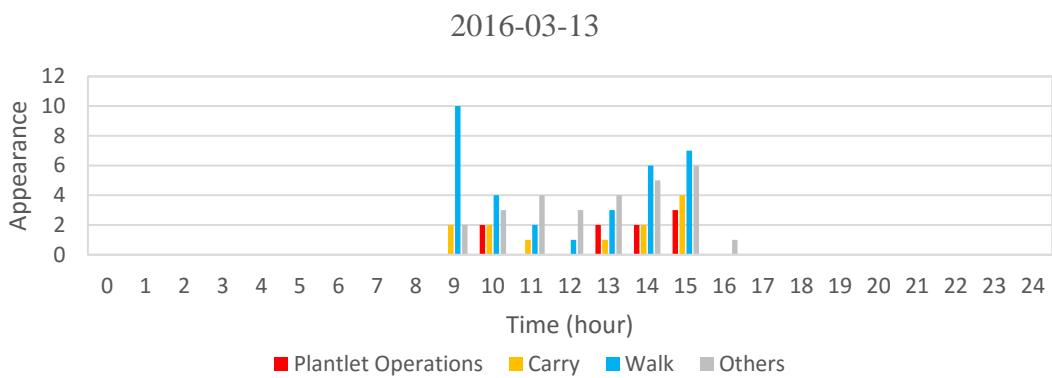
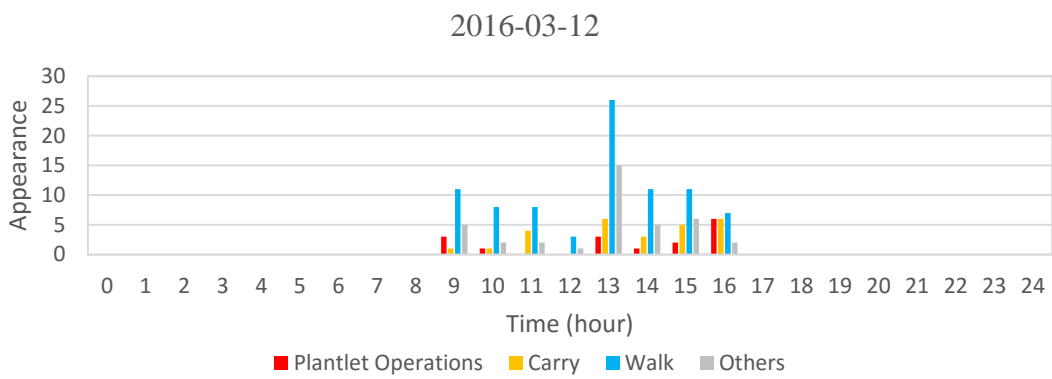
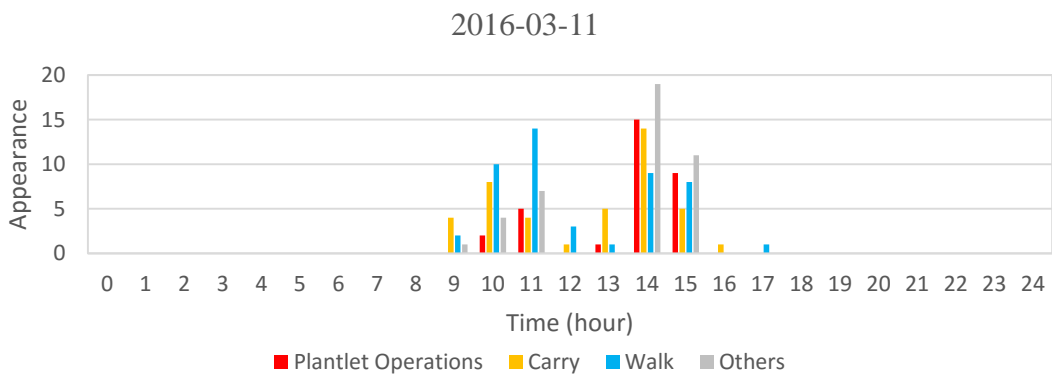


In Figure 4-19 shows the weekly summary of Zhi Chen farm, the same x-axis and the y-axis as Figure 4-18. From the figure, we can see that the activities in 2016-03-11 and 2016-03-12 are much more than the other days. The information can be also acquired in the daily summary in Figure 4-20. We can see the activities of plantlets operations and the carrying plantlets in these two days. When reviewing the recording by hands, we can see the farmers in these two days are packaging the plantlets and send them for delivery. This matches the trend our system records.

Looking back into Figure 4-20, the activities at noon of each day are all lower than in the morning and in the afternoon. This tells us the resting time of the Zhi Chan farm starts from 12 PM to 14 PM. If that day is busy, the resting time will be shorten an hour. We can also find out that sometimes the farmers starts to work in the afternoon instead of starting from the morning in 2016-03-15.



(Continued)



(Continued)

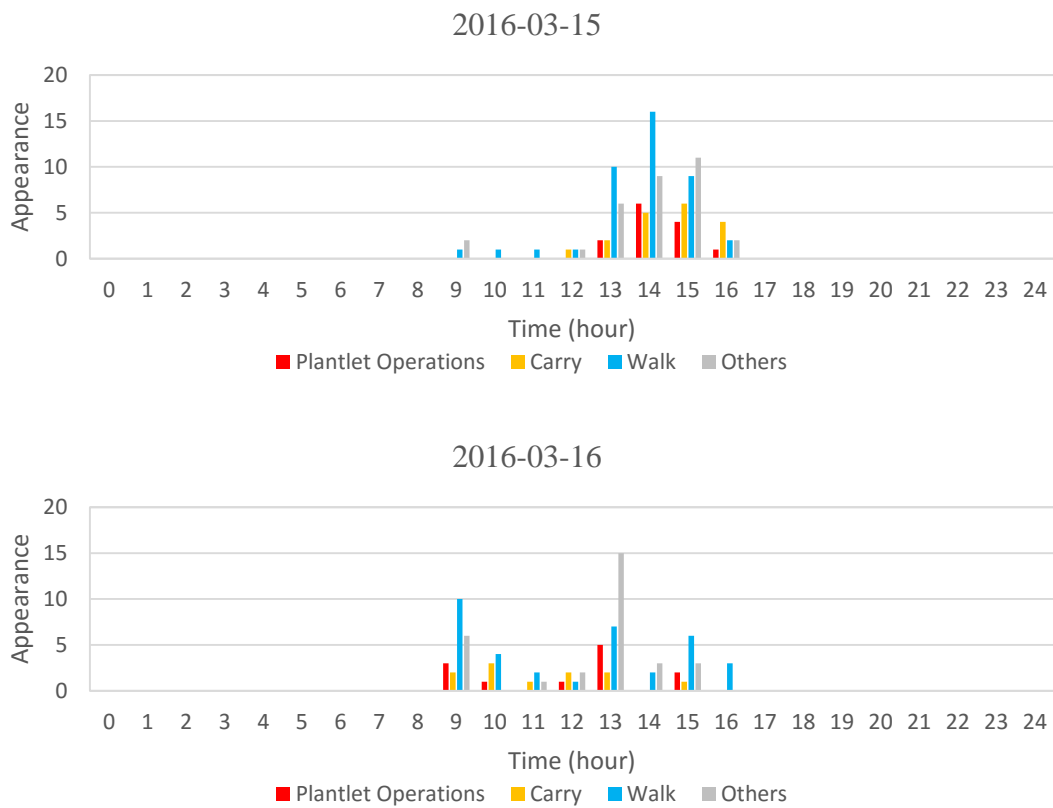


Figure 4-20 Daily summary from 2016-03-10 to 2016-03-16 of Zhi Chen farm

Our system not only able to produce daily and weekly summaries but also able to produce geometry related information. The information include the path usage analysis and most stay positions of a place.

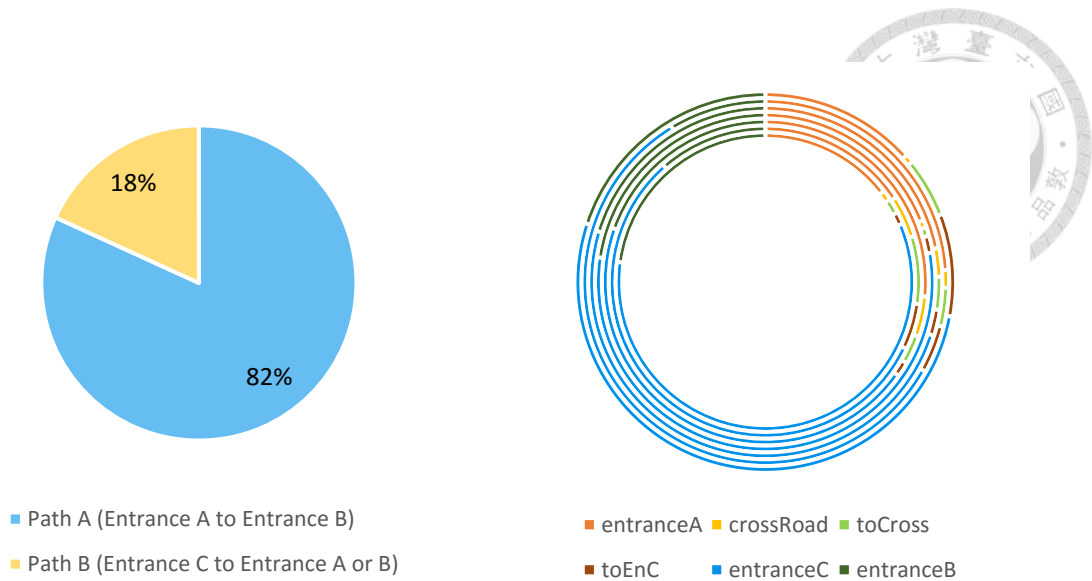


Figure 4-21 Path usage (left) and the most stay positions (right) of the Zhi Chen farm

From the “path usage” pie chart in Figure 4-21, we can see that most of the farmers here walks through path A more often than in path B. We can conclude that path A is the main route of the warehouse that leads to the other sections of the farm. From the “most stay position” pie chart in Figure 4-21, most stay position is entrance C. Entrance C is the entrance leads to another section of the warehouse and is located at the center of the warehouse (Figure 4-22). We can know that the working section is located at entrance C. The second and the third most stay position are entrance A and entrance B, which indicates that path A is the main route of the warehouse and these two positions are the second popular working section in the warehouse from the scene of the camera.



Figure 4-22 Pre-defined map points of Zhi Chen farm

3rd Vegetable packaging factory in Taoyuan County



In the vegetable packaging factory, we get similar results either.

2016-04-27 (first day data of packaging factory)

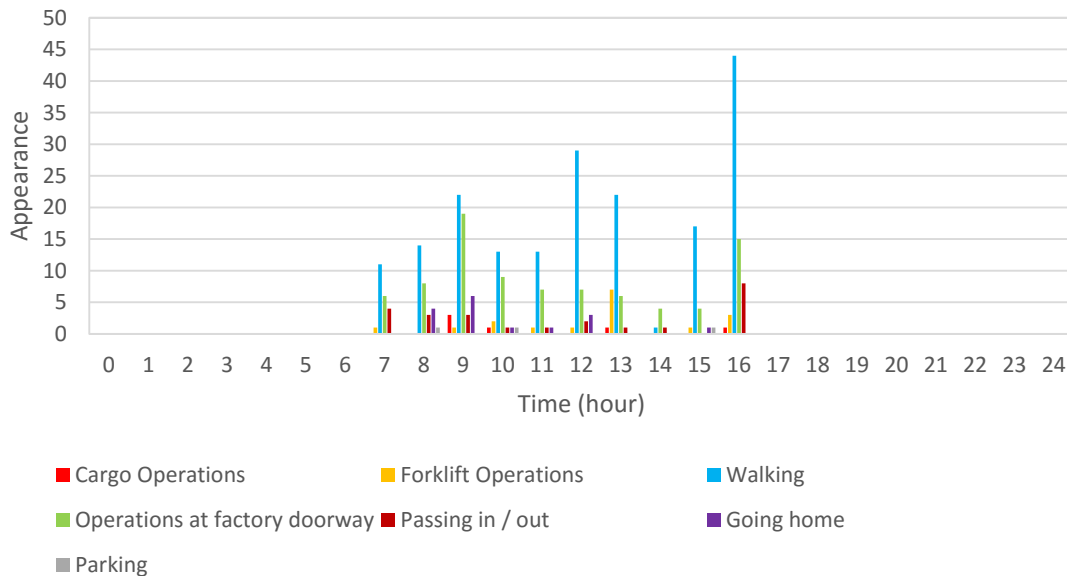


Figure 4-23 The daily result from 3rd vegetable packaging factory of the first day data

Figure 4-23 is the first day data of the 3rd vegetable packaging factory, the y-axis is the appearance of any moving object and the x-axis is the time of 24 hours of a day. We change to detect any moving object instead of the human being because most of the moving objects here are transportations. The system is record from 7 AM to 5 PM. We can see that there is no significant resting time in this factory. However, around 2 PM the activities are less than other time, we may conclude 2 PM might be the resting time. It is quite reasonable that the temperature is the hottest in 2 PM of a day, and it is better to rest instead of working.

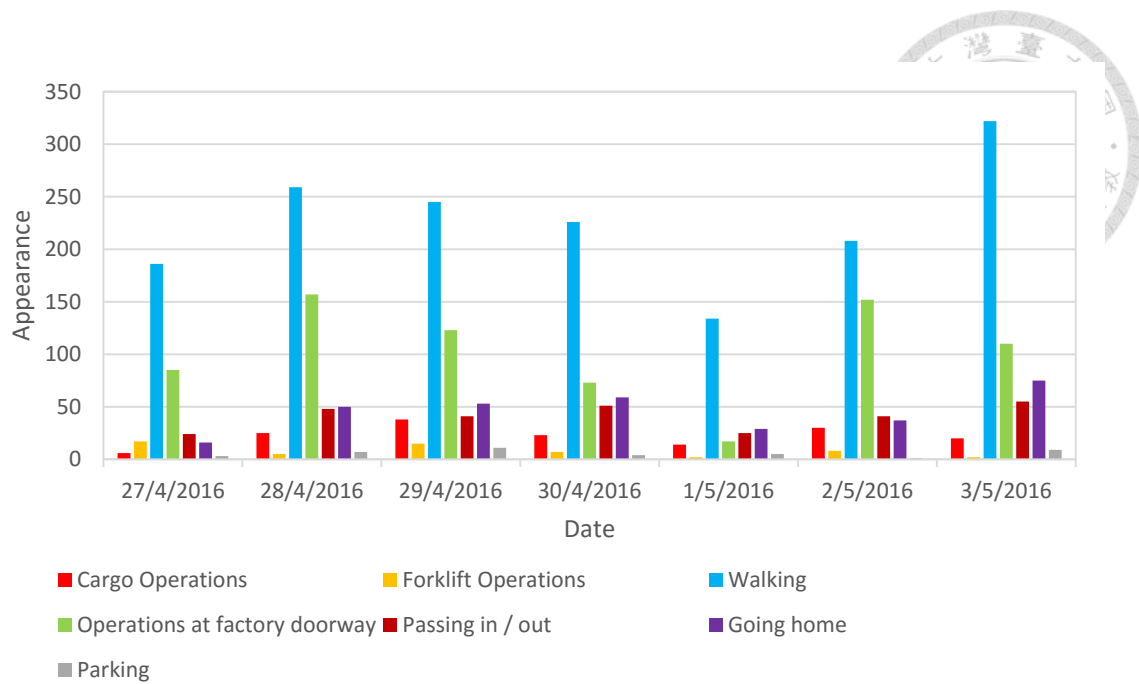


Figure 4-24 Weekly summary of 3rd vegetable packaging factory

In Figure 4-24 shows the weekly summary of the 3rd vegetable packaging factory.

From the figure, we can see that the activities in 2016-05-01 is less than the other days.

The information can be also acquired in the daily summary in Figure 4-25. We can see

the activities happening in this day is just like the other days but with less frequency. If

we check the date on a calendar, we can find out May 1 is Sunday, which is quite

reasonable with less activities.

Looking back into Figure 4-25, the activities at noon of each day are all lower than

in the morning and in the afternoon. However, the specific time of resting is not the same

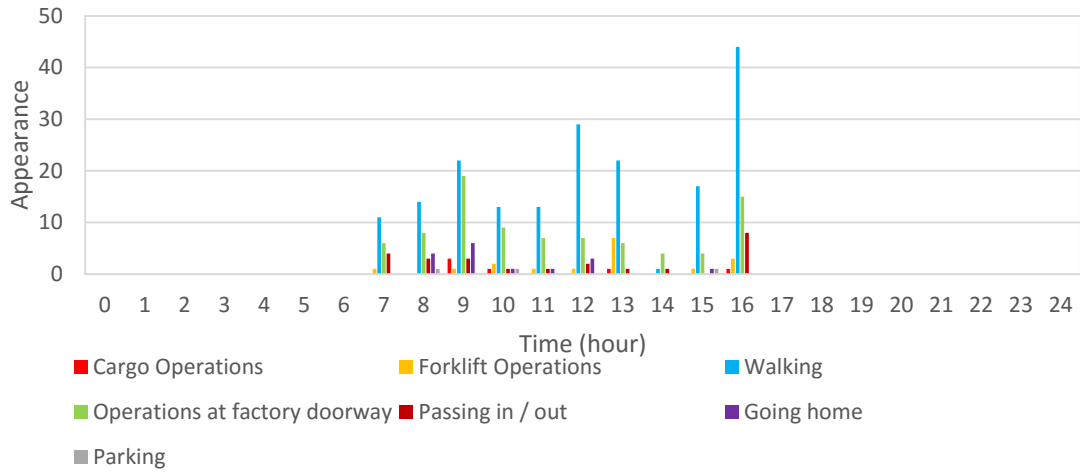
each day, and depends on how busy they are in that day. If that day is busy, the resting

time will be shorten or removed. We can also know the peak of the working time and

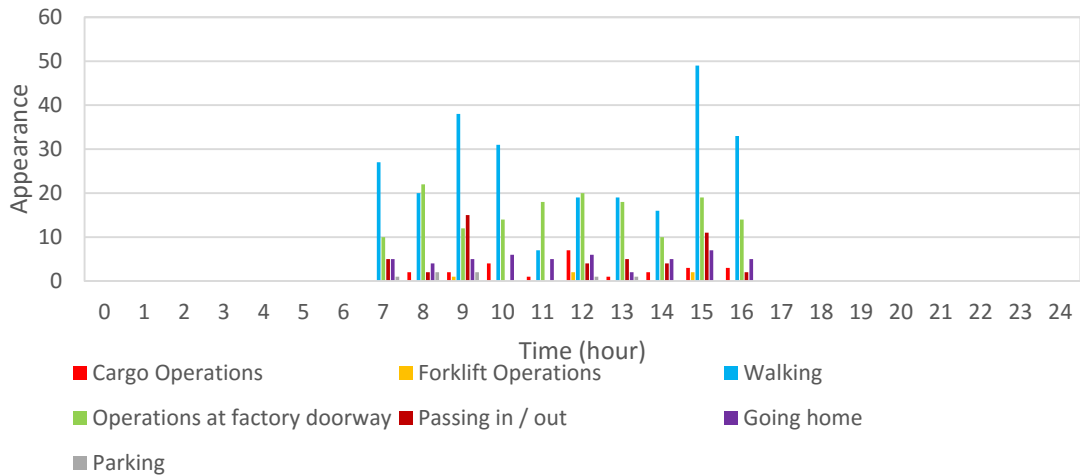
working date by comparing data of 2016-05-01 with other days in daily and weekly summaries.



2016-04-27



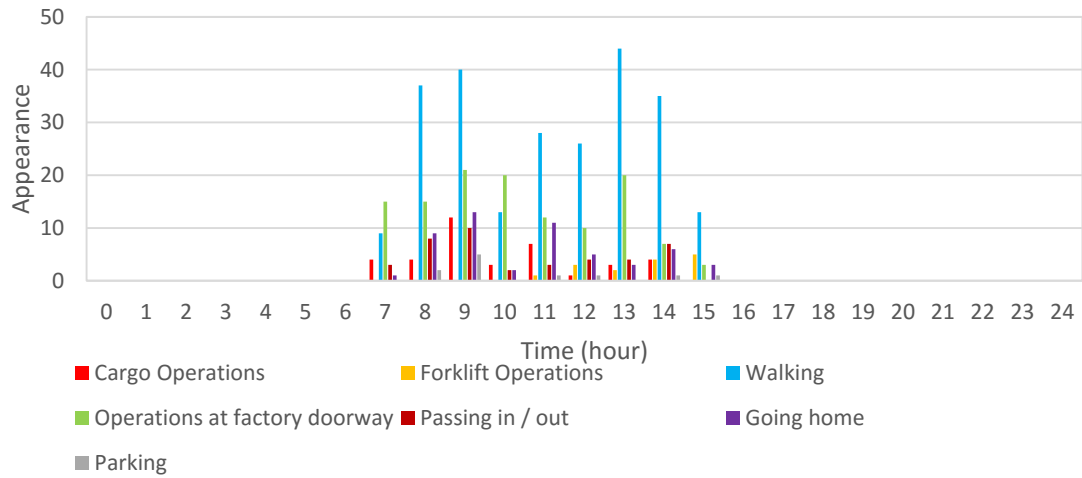
2016-04-28



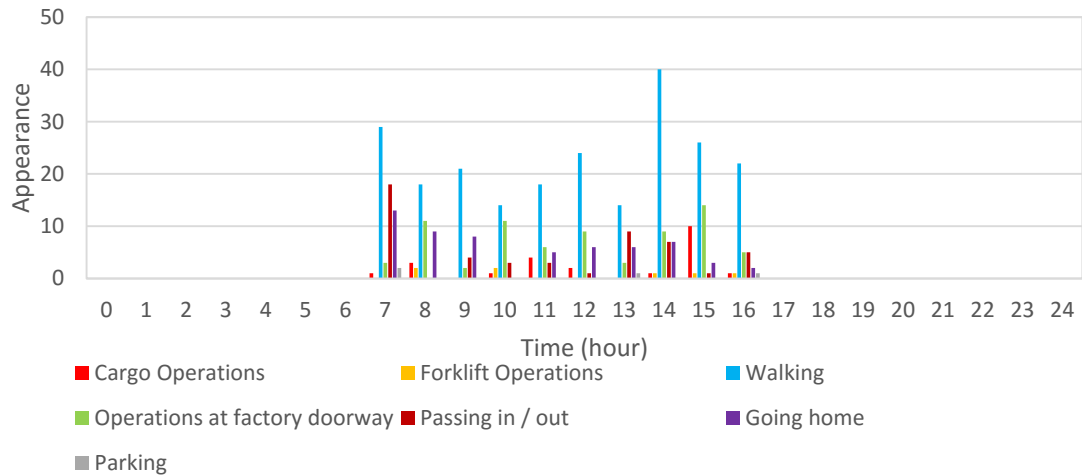
(Continued)



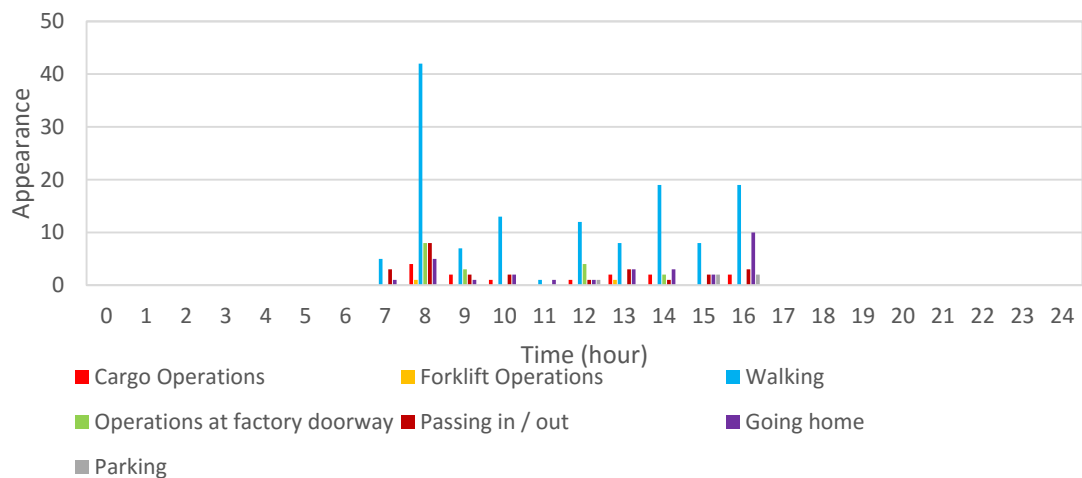
2016-04-29



2016-04-30



2016-05-01



(Continued)

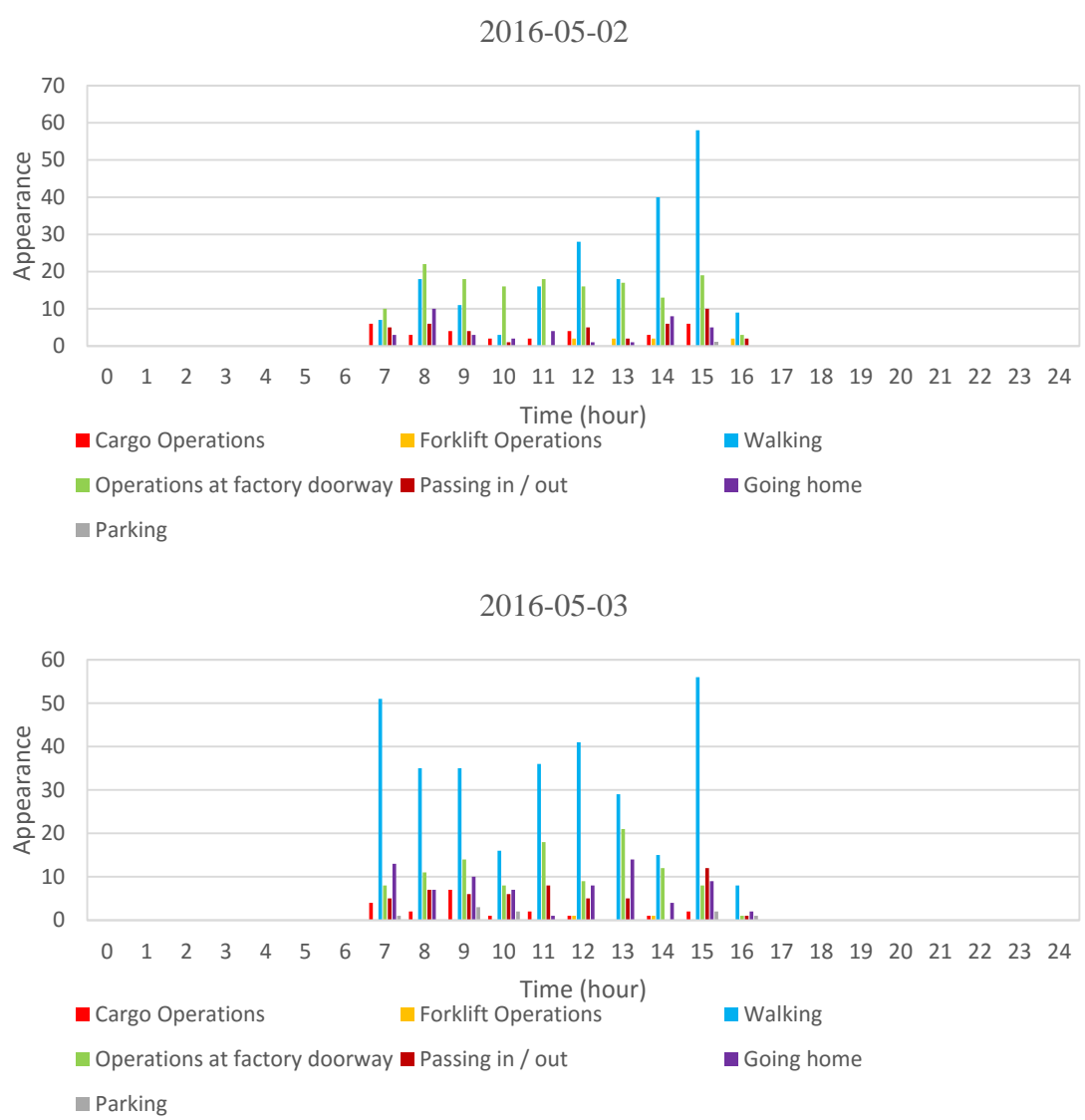


Figure 4-25 Daily summary from 2016-04-27 to 2016-05-03 of 3rd vegetable packaging factory

Because the scenery in the 3rd vegetable packaging factory is a clearing, producing the path usage analysis is quite meaning less. Instead, we have “most stay position” analysis to show that where most of the people stay under or pass through the eye of the camera.

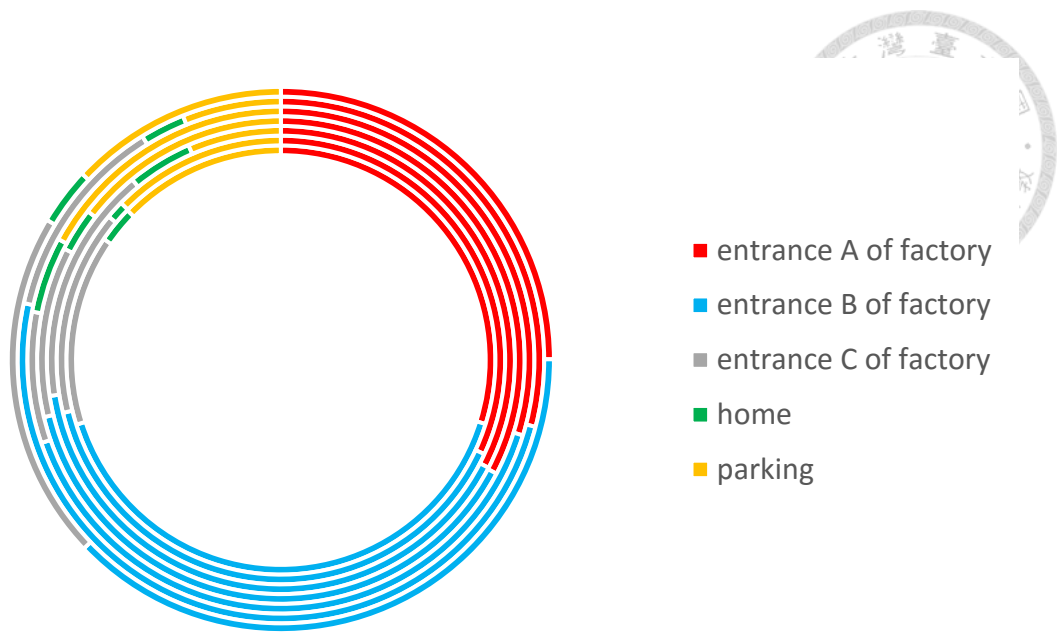


Figure 4-26 Most stay positions of 3rd vegetable packaging factory

From the “most stay position” pie chart in Figure 4-26, most stay position is entrance A of the factory. The represented map points are shown in Figure 4-27. The second and the third most stay point are also the entrances of the factory. This matches the truth working patterns of the factory. Most of the operations such as vegetables loading, vegetable carrying, are done in front of the factory doorway. With our system, we can even tell the users which section of the factory doorway is the most popular staying position.



Figure 4-27 Pre-defined map points of 3rd vegetable packaging factory

Outside the Tomatake Hall

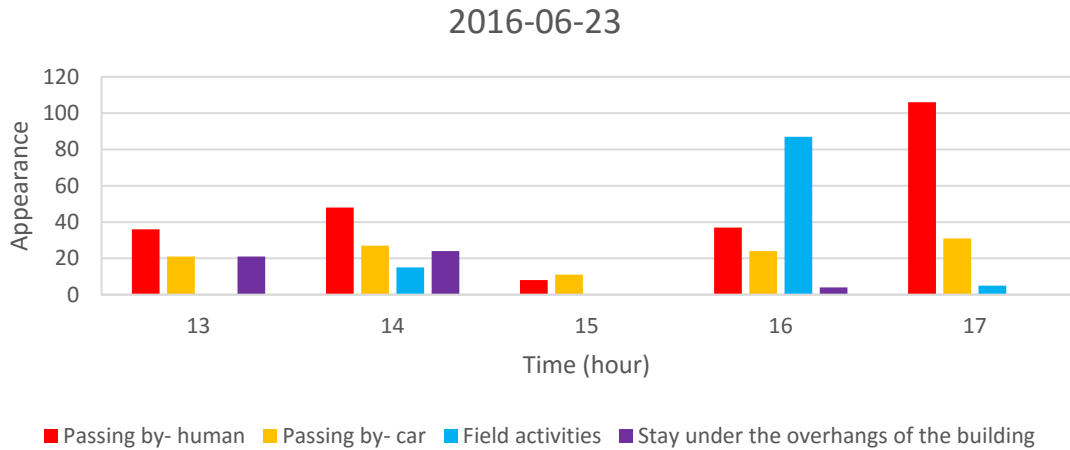


Figure 4-28 Daily summary outside of the Tomatake Hall

Figure 4-28 is the first day of the outside of the Tomatake Hall in NTU, the y-axis is the appearance of any moving object and the x-axis is the time of 24 hours of a day. We do a little traffic-monitoring test here to prove our system is fully environment adaptable. The system is record from 13 PM to 18 PM. The road here is a minor path near the back door of the NTU. The result is surprising that even this minor path in NTU still have hundreds of usage in one single afternoon. We can found out that this road is more suitable for human passing by than car passing by according to the bar chart. We also try to look and monitor at the possible staying position beside the path to see if there is any related, including the sand box, building path (field activities), and the overhangs of the building across from the Tomatake Hall.

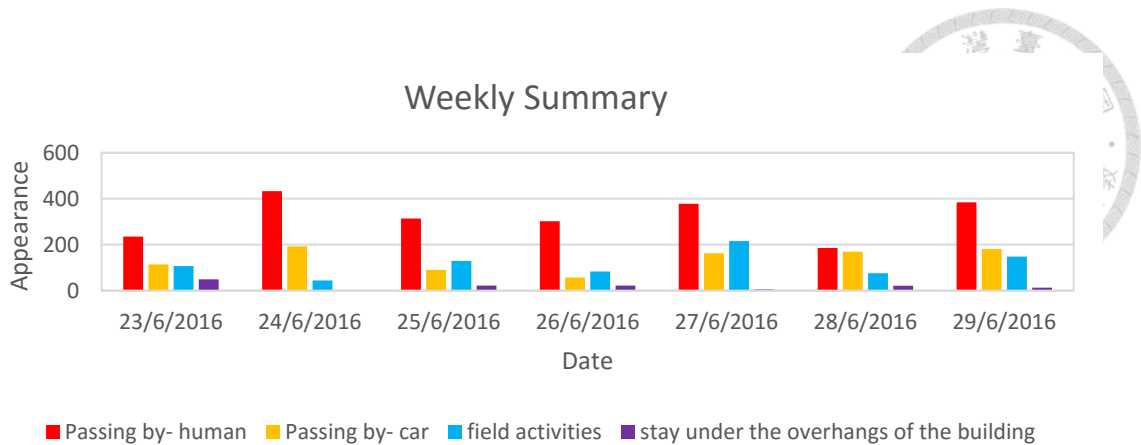
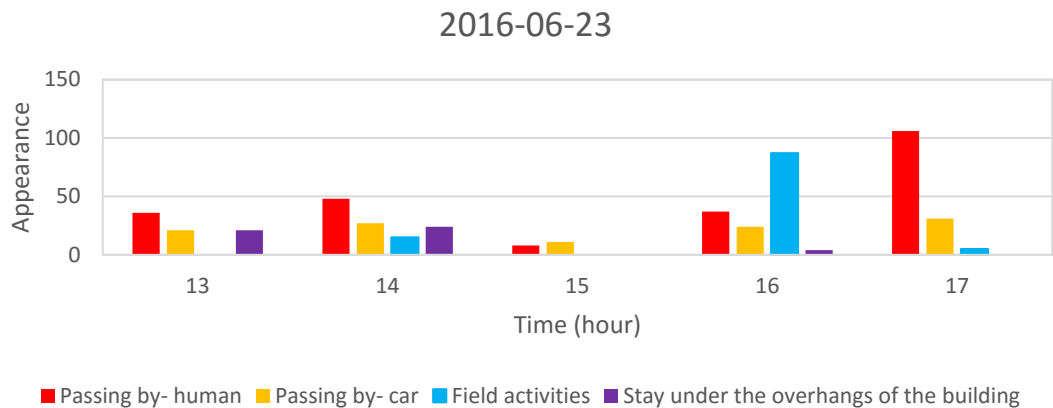
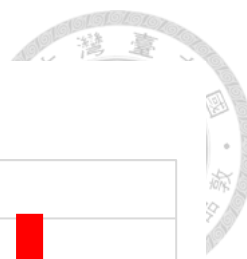


Figure 4-29 Weekly summary outside of the Tomatake Hall

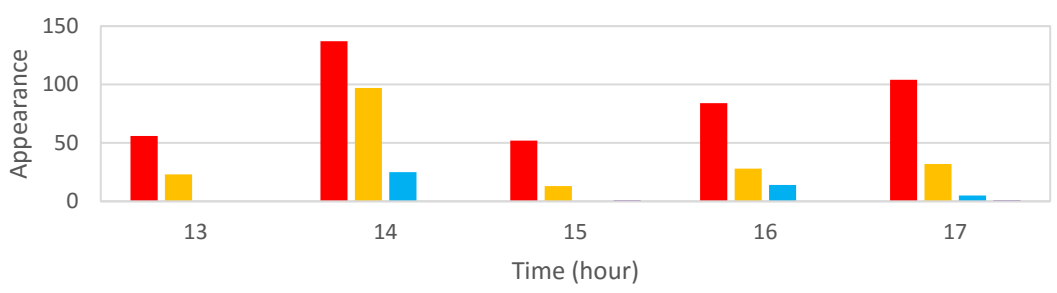
Figure 4-29 is the weekly summary result outside of the Tomatake Hall. The traffic here is very stable with the average usage of 400 passed-by per afternoon. We can say that the passed-by humans are more than the passed-by cars. Something interesting is that although June 25 and 26 are Saturday and Sunday, the activities here are still about the average. Normally, the path usage should be far lower than weekdays because students here do not require taking classes. Below Figure 4-30 is the daily summaries of a week, the figure might contains more usable information for us to explain.



(Continued)

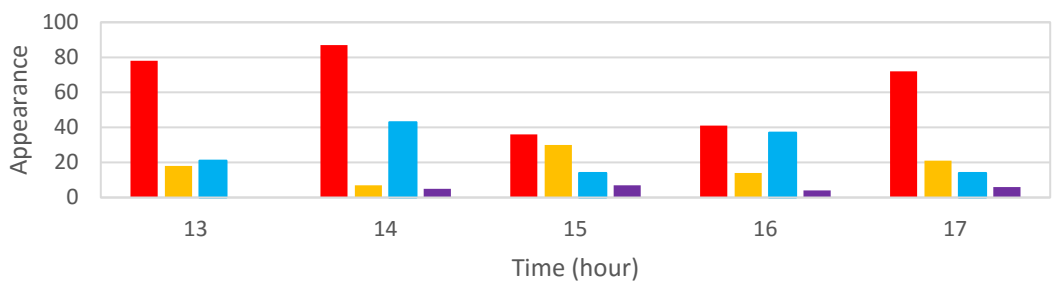


2016-06-24



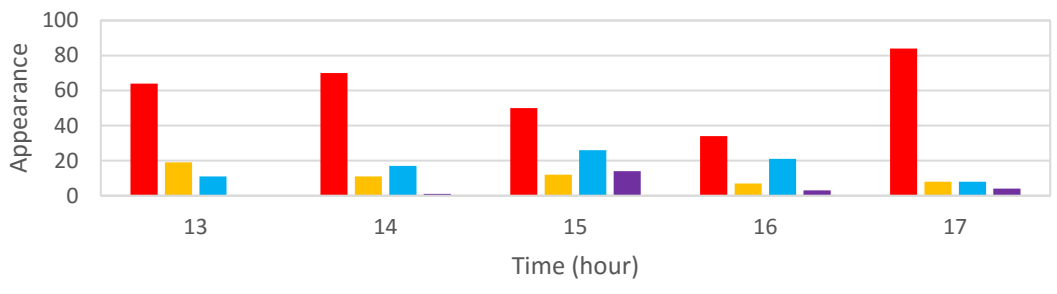
■ Passing by- human ■ Passing by- car ■ Field activities ■ Stay under the overhangs of the building

2016-06-25



■ Passing by- human ■ Passing by- car ■ Field activities ■ Stay under the overhangs of the building

2016-06-26



■ Passing by- human ■ Passing by- car ■ Field activities ■ Stay under the overhangs of the building

(Continued)

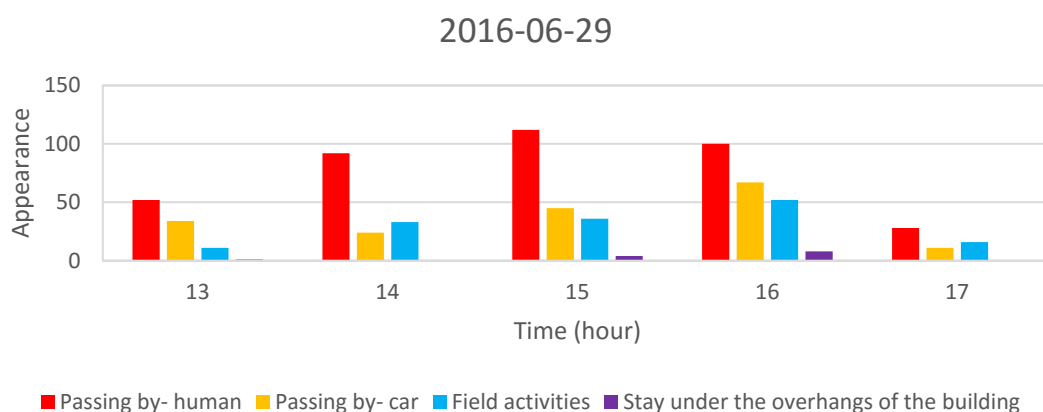
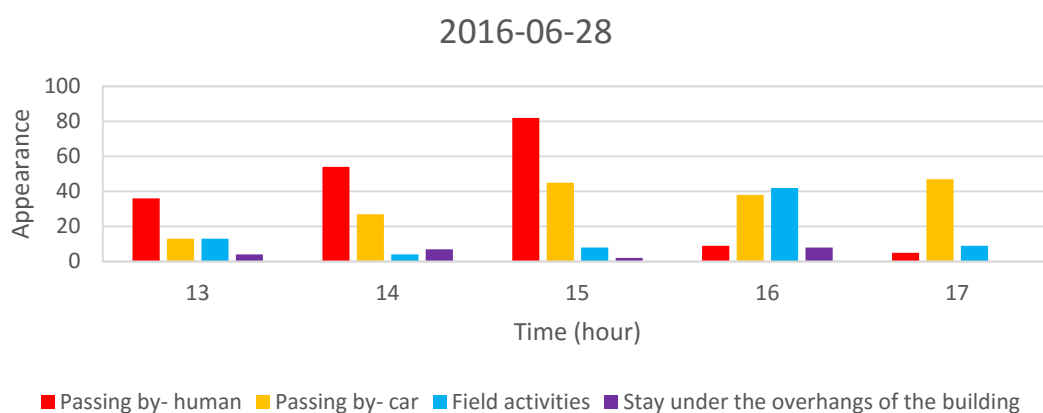
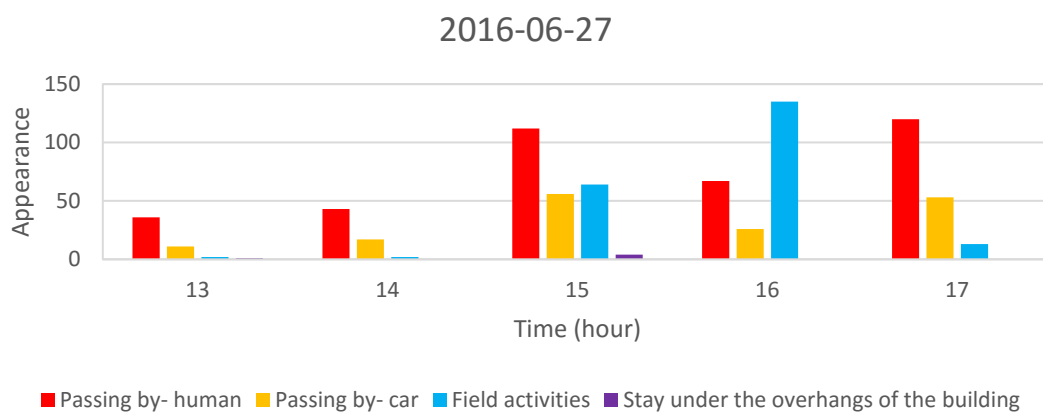
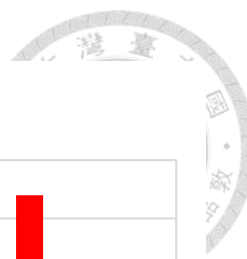
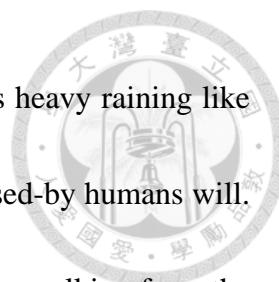


Figure 4-30 Daily summary from 2016-06-23 to 2016-06-29 of outside the Tomatake

Hall

The daily summaries from June 23 to 29 outside of the Tomatake Hall does not seem very pattern, but the activities here are quite repeatedly. Normally, the passed-by humans



are much more than the passed-by cars. If the weather in that day is heavy raining like June 28 around 16 PM, the passed-by cars will be more than the passed-by humans will.

Note that field activities here can be people playing in the sandbox or walking from the building across the Tomatake Hall to the road and vice versa.

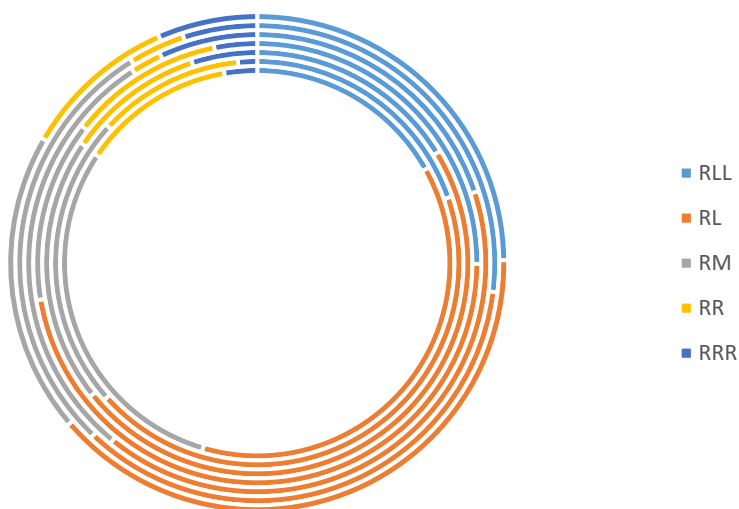


Figure 4-31 Most stay position of the road outside of the Tomatake Hall

Figure 4-31 shows the most stay position among the path. Interestingly, the most staying position of the road is the spot nearest to the entrance of the building across the Tomatake Hall (Small Road Left) (Figure 4-32). If we dig into the recorded video, we can see that the spot is a great position for people to have a small chat. This spot is shadowed under a tree and is near the entrance of the building. When people come out of the building, this spot will naturally become a suitable meeting spot for bikers and walkers. Our system is able to provide interesting information that normal surveillance system cannot.

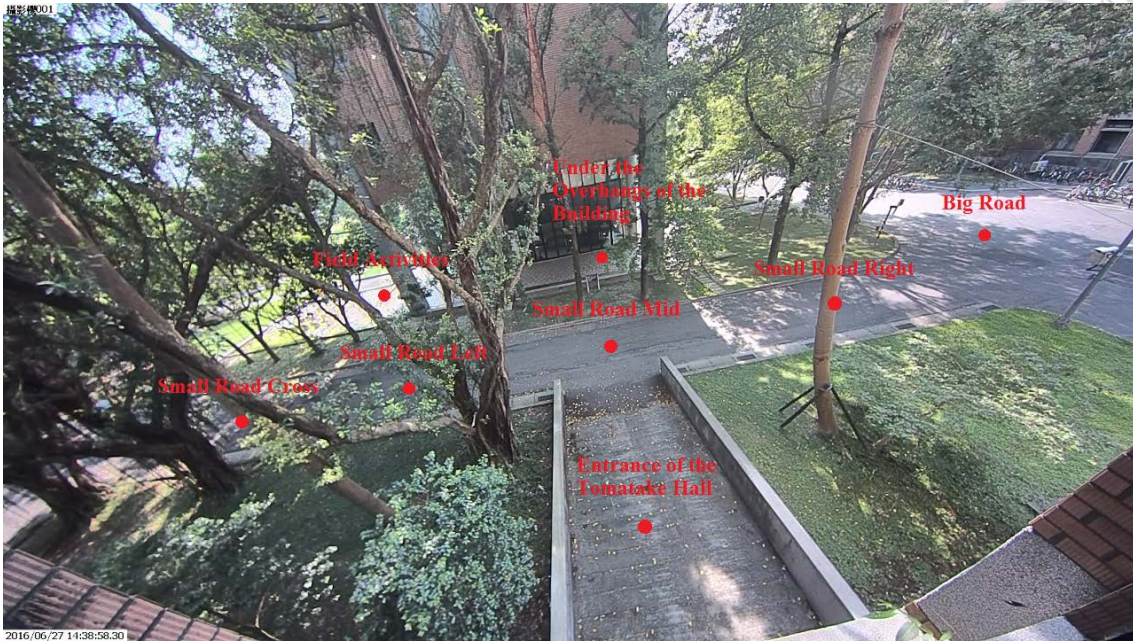


Figure 4-32 Pre-defined map points outside of the Tomatake Hall

Plaza in front of the Dept. of BIME

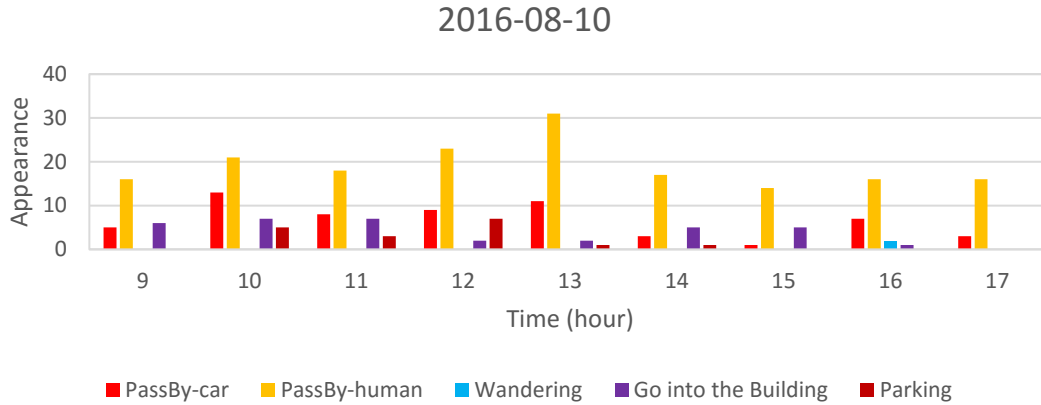
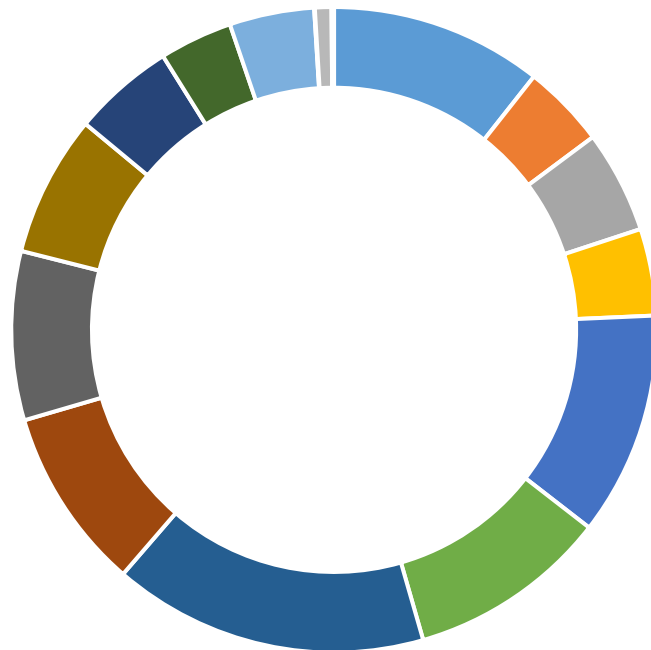


Figure 4-33 Daily summary of the plaza in front of the Dept. of BIME (2016-08-10)

This is a short experiment of the daily summary of the plaza in front of the Dept. of BIME. We monitored the entrance of the department to observe the path usage. The result can be seen in Figure 4-33. Surprisingly, not so much people and cars passed by compare to previous cases. When we review some data by the Object View Manager, we found out that most of the people passed here are the professors and student of the Dept. of BIME. This meets the result of the most stay position in Figure 4-34. The most high-frequently used spot are parkingLot, and the spots among entrance to stairway. Figure 4-35 shows the pre-defined map points in image. The pre-defined points in this case is quite many so it is detailed listed in Table 3-15.



Most Stay Position



- parkingFront
- parkingLot
- stairway
- stairFront
- roadEntranceA
- A-B3
- A-B2
- roadEntranceB
- B-C1
- B-C2
- B-C3
- B-C4
- roadEntranceC
- roadEntranceC-2
- hill1
- hill2
- hill3

Figure 4-34 Most Stay Position of the plaza in front of the Dept. of BIME

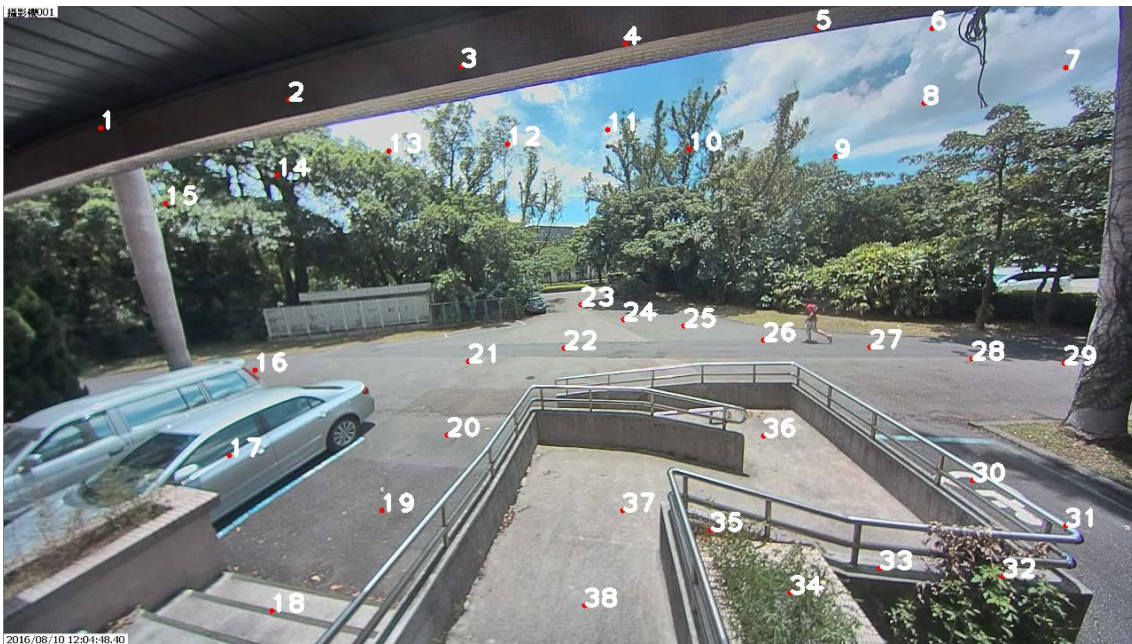


Figure 4-35 Pre-defined map points of the plaza in front of the Dept. of BIME



4.3 Software Performance

Our system uses an ultra-wide FOV camera instead of using panorama images. Although panorama image has the advantage of capture every movement in a place when the camera is placed at the center of the area, the computing time of stitching images into panorama takes too long. By replacing the system to a corner of an area, we are also able to get most of the information of a place. We are also able to detect objects with higher velocities. We also use OpenCL to speed up the whole process. Comparing to the previous system, the processing time is significant increased. The saved time can be used to add more processing technics or add a new camera in another direction to make a surveillance network.

Table 4-19 The performance comparison between the old system and the new system

	FPS
Old system (Panorama stitching)	4 (360 degree)
New system (ultra-wide FOV)	60 (135 degree)

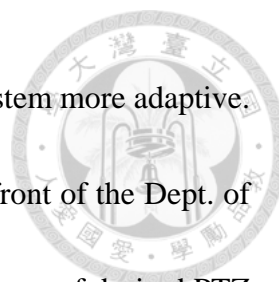
Chapter 5 Conclusions and Suggestions



5.1 Conclusions

We successfully developed a cooperative surveillance system to perform an all-in-one task from observation to final working pattern results of a place. The system is able to analyze the daily and weekly behavioral summaries instead of manually monitoring by human eyes, and provide a nice GUI for users to look up every tracked object detail. We also provide customizations for users to meet their needs. With the combination of different kind of cameras, we are able to find the maximum information.

To achieve the goals of our objectives, we designed five different experiments to approach. The first experiment in the NTU farm tested the simple case of the static object tracking. We verified that if the object is close enough to the camera, a dynamic Euclidean Distance threshold is required to make sure the object will not lose track when certain camera-viewing angle is set. This solves the problem when the object is too close to the camera, any move of the object might exceeds the threshold. By the second experiment in the Zhi Chen farm, we set up a brand new Custom Define System in order to fit different environment. We can quickly setup the system by a simple click-and-type and does not need to recompile our program when the experimental site changes. In the third experiment in 3rd Vegetable packaging factory, we verified our system is able to easily setup and customize to fit the environment with the previous changes. We added some



more pre-defined feature types to make the Custom Define Rules System more adaptive.

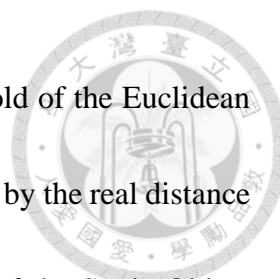
With the last two experiments (outside the Tomatake Hall, plaza in front of the Dept. of BIME), we added a PTZ self-tracking algorithm to increase the accuracy of desired PTZ images of our system by letting the PTZ to research the tracked object when the static camera send a coordinate to the PTZ. We improve our system's stability and change the data and system structure. The system structure is redesigned to be more "cooperative" by using programming technics- parallel programming. The data structure is rewritten to use Windows File Manager instead of creating our own database. We use system's internal category functions to quick search any tracked object we want. Comparing to the old system, we have some changes and new features:

1. Using ultra-wide FOV images instead of panorama images by changing the webcam arrays to Sony network camera.

We removed the panorama imaging system. The computing speed has increase 15 times comparing to the old system. Since the computing time has significantly decrease, the saved time can be used to add more functions or to install a higher frame imaging hardware to perform real-time tasks.

2. Improved object tracking methods

We improved the Gaussian Mixture Model background subtraction with multi-resolution AND logic process. This change is able to remove some noise produced



from the background subtraction. We also use a dynamic threshold of the Euclidean distance to solve the problem of the non-constant threshold cause by the real distance between the object and the camera lens. With the combination of the Static Object Detection, we are able to keep tracking of the object that temporarily static.

3. Custom Define Rules System

We provide users to customize their own analysis results with our unique module system without recompiling the program. By using the existing raw features from the hardware as basic rules, the users are able to create their own custom define rules. Moreover, the created rules can be used as basic rules to create a second layer custom define rules. The cooperative surveillance system is a flexible and self-adaptive tool for the owners and the managers to run in their facilities.

4. Working Pattern Analysis

We are able to use several features to represent the working patterns we assigned to the system. The system fetches the rules we created from the Custom Define Rules system to create daily and weekly summaries. We let the surveillance system to do the long-time observations and analyze for us to replace manual monitoring.



5. Friendly Graphic User Interface

The cooperative surveillance system provides a Video Live View to monitor the cameras in real-time, and an Object View Manager to check every tracked detail with individual analysis results and captured images. The system is not only a researcher purpose but also an amiable product.

5.2 Suggestions

1. In order to remove the noise and the failure of the voting method, we suggest stabilize the tracking results to avoid disconnected trajectories. Improving the core algorithm of the background subtraction will be a great choice.
2. It is better to find more representative module features and create a LabVIEW like system for users to create more custom rules with ease, especially creating a drag-n-drop GUI for fast prototyping. The modules of the Custom Define Rules System can be used to create multiple-layer working pattern analysis for further data mining.
3. The weight of the system is quite heavy, it is necessarily to make the system lighter to make it easier to install and provide a better mobility. The current PTZ can be replaced by a smaller and newer model, and the aluminum frame can be replaced by other lighter materials that are strengthen enough.

References



余世忠。2012。主從式影像監視系統之研製與生態監測應用。碩士論文。台北：台灣大學生物產業機電工程學系

ACTi. 2015. CAM-6630 | ACTi Coporation. Available at:

<http://www.acti.com/product/CAM-6630>. Accessed 14 October 2015

Ahonen, T., R. Virrankoski, and M. Elmusrati. 2008. Greenhouse Monitoring with Wireless Sensor Network. In *Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*.

Aleksander, I., and H. Morton. 1995. *An Introduction to Neural Computing* 2nd ed.

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp. 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on* 50(2):174-188.

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool. 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding* 110(3):346-359.

Bin, W., and P. Dudek. 2014. A Fast Self-Tuning Background Subtraction Algorithm. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*.

Bohyung, H., Z. Ying, D. Comaniciu, and L. Davis. 2005. Kernel-based Bayesian filtering for object tracking. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*.



Boiman, O., and M. Irani. 2007. Detecting Irregularities in Images and in Video. *Int. J. Comput. Vision* 74(1):17-31.

Chang, R., C. Teck Wee, K. Leman, W. Hee Lin, and Z. Jie. 2013. Automatic cooperative camera system for real-time bag detection in visual surveillance. In *Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference on*.

Chen, C.-H., Y. Yao, D. Page, B. Abidi, A. Koschan, and M. Abidi. 2008. Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking. *Circuits and Systems for Video Technology, IEEE Transactions on* 18(8):1052-1063.

Chong, D., S. Bi, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury. 2012. Collaborative Sensing in a Distributed PTZ Camera Network. *Image Processing, IEEE Transactions on* 21(7):3282-3295.

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

Cover, T., and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1):21-27.

Dalal, N., and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*.



Diebold, F. 2007. *Elements of Forecasting*. 1 ed. Cengage Learning.

Digia. 2015. Qt. Available at: <http://www.qt.io>. Accessed 14 October 2015

Doucet, A., N. de Freitas, and N. Gordon. 2001. An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice*, 3-14. A. Doucet, N. de Freitas, and N. Gordon, eds: Springer New York.

Feichtinger, H. G., and T. Strohmer. 1998. *Gabor Analysis and Algorithms*. Birkhäuser.

Feizi, A., A. Aghagolzadeh, and H. Seyedarabi. 2013. Using optical flow and spectral clustering for behavior recognition and detection of anomalous behaviors. In *Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on*.

Fischler, M. A., and R. C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24(6):381-395.

Fletcher, R. 1987. *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience.

Foroughi, H., B. S. Aski, and H. Pourreza. 2008. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*.

Ghidoni, S., A. Pretto, and E. Menegatti. 2010. Cooperative tracking of moving objects and face detection with a dual camera sensor. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*.

Hashimoto, Y., D. Arita, A. Shimda, T. Okayasu, H. Uchiyama, and R.-i. Taniguchi. 2016.

Farmer Position Estimation in a Tomato Plant Green House with Smart Devices.

Hecht-Nielsen, R. 1989. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on.*

Jingqing, W., and Z. Xi. 2015. Cooperative MIMO-OFDM based multi-hop 3D clustered wireless camera sensor networks. In *Wireless Communications and Networking Conference (WCNC), 2015 IEEE.*

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82(1):35-45.

Kang, S., J. Paika, A. Koschan, B. Abidi, and M. A. Abidi. 2003. Real-time video tracking using PTZ cameras.

Li, Y., C. Xia, and J. Lee. 2009. Vision-based pest detection and automatic spray of greenhouse plant. In *2009 IEEE International Symposium on Industrial Electronics.*

Michelsoni, C., B. Rinner, and G. L. Foresti. 2010. Video Analysis in Pan-Tilt-Zoom Camera Networks. *Signal Processing Magazine, IEEE* 27(5):78-90.

Popoola, O. P., and W. Kejun. 2012. Video-Based Abnormal Human Behavior Recognition—A Review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 42(6):865-878.



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

Russell, S., and P. Norvig. 2009. *Artificial Intelligence: A Modern Approach*. 3 ed.

Saad, S. M., L. M. Kamarudin, K. Kamarudin, W. M. Nooriman, S. M. Mamduh, A. Zakaria, A. Y. M. Shakaff, and M. N. Jaafar. 2014. A real-time greenhouse monitoring system for mango with Wireless Sensor Network (WSN). In *Electronic Design (ICED), 2014 2nd International Conference on*.

San Miguel, J. C., and J. M. Martinez. 2008. Robust Unattended and Stolen Object Detection by Fusing Simple Algorithms. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*.

Sony. 2015. SNC EB-630 (SNCEB630): Product Overview: Other: Sony Professional. Available at: <http://www.pro.sony.eu/pro/lang/en/eu/product/video-security-ip-cameras-fixed/snc-eb630/overview/>. Accessed 14 October 2015

Stillman, S. T., R. Tanawongsuwan, and I. A. Essa. 1998. A System for Tracking and Recognizing Multiple People with Multiple Cameras Georgia Institute of Technology

Su Jeong, Y., P. H. Truong, J. Sang Hoon, L. Sang Moo, L. Chang Eun, and C. Young Jo. 2014. A cooperative multi-camera system for tracking a fast moving object. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*,



2014 IEEE 4th Annual International Conference on.

Suykens, J. A. K., and J. Vandewalle. 1999. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9(3):293-300.

Tamaki, M., M. Hirata, N. Taniyai, M. Tamaki, T. Akachi, U. Takae, H. Sunagawa, H. Inoue, A. Kaiho, and S. Tamaki. 2015. Daily CO₂ changes monitored by an environment sensor unit in a greenhouse with bitter gourd. In *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*.

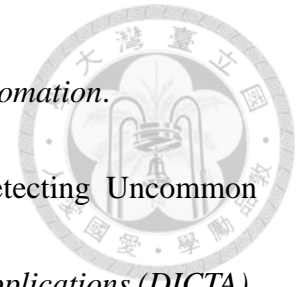
Tao, X., and G. Shaogang. 2008. Video Behavior Profiling for Anomaly Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30(5):893-908.

Theia. 2015. SY125/MY125 Ultra wide- Theia Technologies. Available at: <http://www.theiatech.com/sy125-ultra-wide/>. Accessed 14 October 2015

Utsumi, A., H. Yamazoe, S. Abe, D. Kanbara, and H. Yamauchi. 2006. Human Behavior Recognition for Daily Task Assistance using Sparse Range Data Observations. In *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on.*

Vapnik, V. N. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc.

Wenjing, Z., and W. Xueqiang. 2014. Design of Greenhouse Temperature Detection System Based on Linear Offset Interference. In *2014 Sixth International*



Conference on Measuring Technology and Mechatronics Automation.

Wiliem, A., V. Madasu, W. Boles, and P. Yarlagadda. 2008. Detecting Uncommon Trajectories. In *Digital Image Computing: Techniques and Applications (DICTA), 2008.*

Wu, X., Y. Ou, H. Qian, and Y. Xu. 2005. A detection system for human abnormal behavior. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on.*

Xia, C., Y. Hwang, D. H. Lee, J. Lee, and M. C. Lee. 2015. Three-dimensional plant leaf mapping and segmentation using kinect camera. In *Society of Instrument and Control Engineers of Japan (SICE), 2015 54th Annual Conference of the.*

Yan, Z., and K. Gu. 2014. The Study on the Techniques on Gun-Dome Camera Cooperative Human-Tracing. In *Semantics, Knowledge and Grids (SKG), 2014 10th International Conference on.*

Yang, L., J. Dickinson, Q. M. J. Wu, and S. Lang. 2007. A fruit recognition method for automatic harvesting. In *Mechatronics and Machine Vision in Practice, 2007. M2VIP 2007. 14th International Conference on.*

Yilmaz, A., O. Javed, and M. Shah. 2006. Object tracking: A survey. *ACM Comput. Surv.* 38(4):13.

Zhang, D., D. Gatica-Perez, S. Bengio, and I. McCowan. 2005. Semi-supervised adapted



HMMs for unusual event detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.*

Zhou, X., R. T. Collins, T. Kanade, and P. Metes. 2003. A master-slave system to acquire biometric imagery of humans at distance. 982467: ACM.

Zhou, Y., S. Yan, and T. S. Huang. 2007. Detecting Anomaly in Videos from Trajectory Similarity Analysis. In *Multimedia and Expo, 2007 IEEE International Conference on.*

Zivkovic, Z. 2004. Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on.*

Zweng, A., and M. Kampel. 2010. Unexpected Human Behavior Recognition in Image Sequences Using Multiple Features. In *Pattern Recognition (ICPR), 2010 20th International Conference on.*