Continuous Analysis of Office Video

Kuangzheng Ye

Master of Science

Artificial Intelligence

School of Informatics

University of Edinburgh

2016

Abstract

This dissertation aims to develop a human behaviour classification algorithm, which is capable of recognising actions of people and reporting abnormal behaviours. Ground truth of the data is obtained by a smart labelling tools exploited in this project. Algorithms related to background subtraction, tracking, background update as well as classification implemented in this project are illustrated in detailed. The classification algorithms that adopted in this project is Hidden Markov Model which perform a satisfactory result with the accuracy of 88.26%. Considering that ground truth contains error label, the accuracy of classification could reach approximately 90%.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor, Bob Fisher, who has strong responsibility, a great insight and expertise. He has been providing me with constructive suggestions and instructive advices during the whole project period. I could not image that anyone else would be more suitable than him in supervising this project.

Furthermore, I would also like to acknowledge a visiting scholar, Edigleison Carvalho, a keen and kindness person who has been patient with me and explained exhaustively about classification technique that advanced the project.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Kuangzheng Ye)

Table of Contents

Chapter 1

Introduction

Elderly people left at home alone may be more likely to encounter unexpected danger, such as accidentally falling and not being able to get back up or suffering from a heart attack and not being able to reach help. To lessen this misfortune, it is better to monitor the elderly person inside the person's own room and to immediately recognise any abnormal behaviour and report it to others. This kind of work can be classified as a human behaviour detection task. The tracking and detection task in computer vision has been popular for many years and has been applied to many areas, such as vehicle tracking to assist the driver [33], intelligent environments to improve living quality [34] [35], site security monitoring [36] or even, the most popular topic nowadays, augmented reality [37] [38].

Tracking and analysing a person's behaviour will be the main task of this project. Due to privacy reasons, it is not possible to use home video data, thus, alternatively, in this project office video data was obtained, after permission from the person being monitored was granted. A normal office environment with no people inside is illustrated in Figure 1.1.

It can be seen from Figure 1.1 that this environment is complex, with many object inside the room. Different to some projects that use experiment data that is well designed with all of the objects in a fixed position and constant light conditions, in this project the object in the room may change position and the light condition is not always constant. Moreover, the people who use the room may be there for a long period of time, potentially becoming part of the background. Additionally, lines in the room are all distorted caused by fish-eye effect by the lens, this makes the prior knowledge of

the room less useful, for example the edge of the door is distorted, therefore, relying on simple methods to eliminate the shape of the door from the background and segment it from the people may not work properly. To address these problems, the algorithms related to the background subtraction, tracking and background updates are merged. A demonstration and experiment will be discussed in the following chapter.

Figure 1.1: Office environment

Chapter 2 of this project will introduce the background and a literature review of this project will be given while chapter 3 describes how the data looks like and labelling tools for acquiring the ground truth data. Chapter 4 will discuss in details about the methodology that implemented in the project. Chapter 5 demonstrates the experiments that evaluate the performance of the methodology and chapter 6 will be a conclusion chapter.

Chapter 2

Background

Human behaviour recognition task contains several different procedures that include: ground-truth data acquisition to obtain useful information from the video sequence, an adaptive background subtraction algorithm to extract the person from a noisy background, a person detection and tracking algorithm to detect and track an individual and an action recognition algorithm to classify human behaviour.

2.1 Review of human behaviour recognition

The most relevant study was proposed by Douglas Ayers and Mubarak Shah, in which they set up video based human behaviour recognition in an office environment [2]. Their system is built with three low-level computer vision techniques that refer to skin detection, tracking, scene change detection and one high-level state machine model. The skin detection is used only in areas that belong to the entrance to initialize the system. Instead of using RBG colour space, they choose to use HSV colour space, with the skin detection technique provided by Kjeldsen and Kender [3]. Skin colour is not common in an office environment, which makes it good feature to track in this scenario. The tracking algorithm in [2] is developed by Fieguth and Terzopoulos [4], which consumes a low computation cost and is capable of tracking two people in the scene, although they do not cope with the occlusion problem. When people have been tracked to move to some interesting area the scene change detection will start to function. It computes the log likelihood to determine whether the pixels are changed in the interesting area, to identify if the person is interacting with the objects. A finite state machine (shown in Figure 2.1) is used in their action recognition algorithm; the state

of the person could only transit from certain states. For example, the person's state cannot transit from ''open cabinet'' to ''pick up phone''. The system they designed was capable of recognizing human behaviour, such as entering the room, leaving the room, using the computer, opening a cabinet and picking up a phone. However, the paper does not state what will happen and how to solve it, if transition outside their state machine occurred. Prior knowledge, related to the layout of the room, was used in the system, but they did not use any machine learning method to make a prediction of the transitions from state to state. One difference between this new project and theirs is that the light conditions in their environment are almost constant and the object position is required to be in the same place, whereas, in my project, the light condition will be affected by the environment, either outside or inside (the light may be turned off from time to time) and objects will sometimes change position, which will require an adaptive background subtraction method.

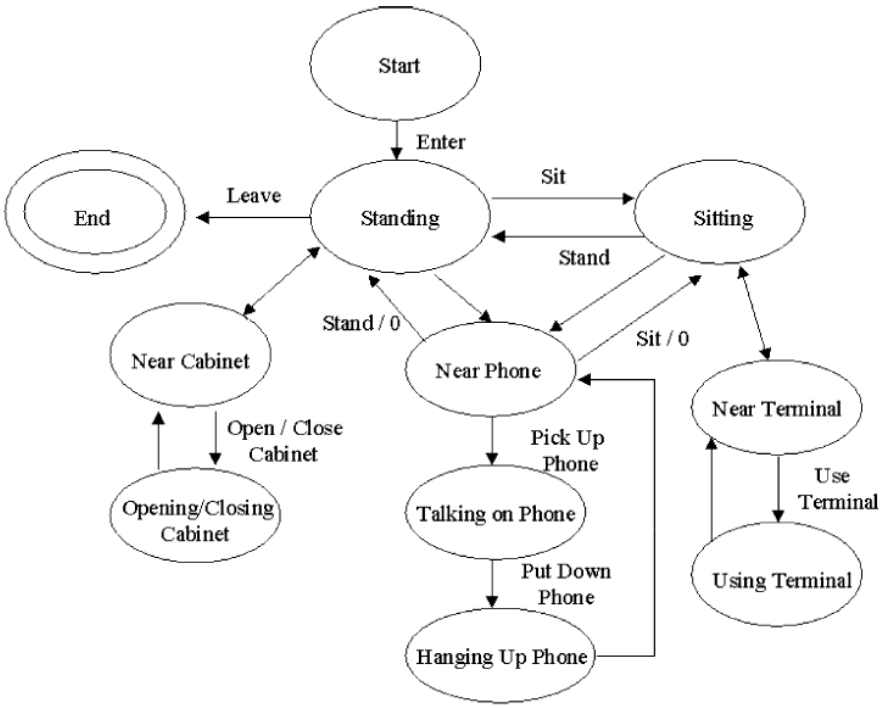


Figure 2.1: State machine model [2]

In another study led by Junji YAMATO [5], they were the first to propose the idea of using Hidden Markov Model in recognizing human behaviour. They apply a bottom up approach and use a mesh feature [6], which is then assigned a symbol that relates to a codeword in the codebook proposed by Vetor Quantization [7]. The feature that

is extracted is fitted into the Hidden Markov Model for each of the four categories. The model that best describes the data from one of the four HMMs will be adopted. The accuracy they could achieve in recognition is higher than 90% and it is said that the accuracy could be improved with more data. A Hidden Markov Model is also adopted in the study led by Ji Tao and Yap-Peng Tan [8]. They use low-level colour features to represent people and then apply these features to a HMM to track people, in order to see who is entering or leaving the room.

2.2 Background Subtraction

In almost all of the computer vision applications, the background subtraction method is the first basic task that needs to be performed. A variety of methods have been proposed in this area, with some even being robust to lighting change, or introducing new objects.

2.2.1 Basic Background Subtraction algorithm

The most common Background subtraction (BS) method is by comparing the difference between the background image and current frame in a video sequence then threshold the difference to determine if a pixel belong to the background or foreground. There are several different ways to measure the difference [9]:

$$d_0 = |I_{s,t} - B_{s,t}| \tag{2.1}$$

$$d_1 = \left|I_{s,t}^R - B_{s,t}^R\right| + \left|I_{s,t}^G - B_{s,t}^G\right| + |I_{s,t}^B - B_{s,t}^B| \tag{2.2}$$

$$d_2 = (I_{s,t}^R - B_{s,t}^R)^2 + (I_{s,t}^G - B_{s,t}^G)^2 + (I_{s,t}^B - B_{s,t}^B)^2 \tag{2.3}$$

$$d_\infty = \max\{\left|I_{s,t}^R - B_{s,t}^R\right|, \left|I_{s,t}^G - B_{s,t}^G\right|, |I_{s,t}^B - B_{s,t}^B|\} \tag{2.4}$$

Where d_0 is for measuring the difference with greyscale, while the others are for measuring the difference for the RGB (could change to the other colour space) image. The background updated rule for the basic BS method is as follows:

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha \cdot I_{s,t} \tag{2.5}$$

Where α is a range value between 0 and 1, which is used to decide how much the current frame pixel goes into the background.

2.2.2 Model background with One Gaussian

The basic background subtraction algorithm cannot deal with such problems as background noise or artefacts and, therefore, scientists seek another way to model each background pixel with a probability model. The higher the probability, the more likely that pixel belongs to the background [9]. In the study from [10], the author models the colour histogram of each pixel with a single Gaussian with full covariance matrix. The distance metric can be as follows:

$$d_G = \frac{1}{2}\log((2\pi)^3|\sum_{s,t}|) + \frac{1}{2}(I_{s,t} - \mu_{s,t})\sum_{s,t}^{-1}(I_{s,t} - \mu_{s,t})^T \quad (2.6)$$

Where $\mu_{s,t}$ and $\sum_{s,t}$ are the mean and covariance matrix at pixel s in time t, respectively,

or using Mahalanobis distance:

$$d_M = (I_{s,t} - \mu_{s,t})\sum_{s,t}^{-1}(I_{s,t} - \mu_{s,t})^T \quad (2.7)$$

The noisy area of an image has larger values than the static area in the covariance matrix, which means the covariance matrix could threshold locally, according to the noise. This enables this model to be more robust than the basic background subtraction methods. The updating rules for this could be:

$$\mu_{s,t+1} = (1 - \alpha) \cdot \mu_{s,t} + \alpha \cdot I_{s,t} \quad (2.8)$$

$$\sum_{s,t+1} = (1 - \alpha) \cdot \sum_{s,t} + \alpha \cdot (I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T \quad (2.9)$$

2.2.3 Gaussian Mixture Model

A single Gaussian Model will be enough for modelling the background, if every pixel is from a unique surface and in the constant lightning, or, if there is only slight change, a Gaussian is still sufficient in the case [11]. However, to handle a background with multiple textures, or if the lightning conditions change, it is better to use a mixture model to model each pixel in the image. Chris Stauffer and W.E.L Grimson used a mixture of 3 Gaussian models in their background modelling method. The probability of a pixel with a colour can be calculated by [9]:

$$\mathrm{P}(I_{s,t}) = \sum_{i=1}^{K} w_{i,s,t} N(\mu_{i,s,t}, \sum_{i,s,t}) \quad (2.10)$$

Where $N(\mu_{i,s,t}, \sum_{i,s,t})$ is the i_{th} Gaussian model while $w_{i,s,t}$ represents its weight. To reduce the computational cost, it is suggested that the covariance matrix could be set as diagonal. The updated rule for Gaussian Mixture Model is as follows [9]:

$$w_{i,s,t} = (1-\alpha) w_{i,s,t-1} + \alpha \quad (2.11)$$

$$\mu_{i,s,t} = (1-\rho) \cdot \mu_{i,s,t-1} + \rho \cdot I_{s,t} \quad (2.12)$$

$$\sigma^2_{i,s,t} = (1-\rho) \cdot \sigma^2_{i,s,t-1} + \rho \cdot d_2(I_{s,t}, \mu_{i,s,t}) \quad (2.13)$$

Where α and ρ are both learning rates, ρ could be defined as $\rho = \alpha N(\mu_{i,s,t}, \sum_{i,s,t})$. The prior weight of the GMM will be adjusted as follow:

$$w_{i,s,t} = (1-\alpha) w_{i,s,t} \quad (2.14)$$

To decide where a pixel value in the image belongs to the background, or foreground, the colour of that pixel is compared with the mean value from every H distribution. If that pixel colour is 2.5 standard deviations away from the mean value, it is said to be the foreground. The H distribution is defined as follows:

$$\mathrm{H} = argmin_h(\sum_{i=1}^{h} w_i > \tau) \quad (2.15)$$

2.2.4 Kernel Density Estimation

Another approach to handle a background that is cluttered and not entirely static (moving tree branches or bushes) has been proposed by Ahmed Elgammal and David Harwood [12]. The model they used is a non-parametric model that allows quick changes in the scene and is very sensitive moving objects. It could also be used to suppress shadows to enhance detection. To define a pixel that is in the foreground, this model uses a kernel based estimation algorithm, as follows:

$$\mathrm{P}(I_{s,t}) = \frac{1}{N} \sum_{i=t-N}^{t-1} K(I_{s,t} - I_{s,i}) \quad (2.16)$$

Here, K is the kernel function and the most typical kernel is a Gaussian. If the estimator has a small value for a pixel, that pixel will be defined as the foreground. The problem

with this algorithm is the memory cost, as an N*size (frame) has to be stored to compute the kernel value.

Other approaches, such as Codebook [13], Eigen background [14], Mean-shift based estimation [15] or modelling each pixel with a Kalman Filter in the background [16], are also possible solutions for background subtraction, or foreground detection works.

2.3 Object Tracking

The object tracking technique has been a topic of heated debate in the research area for many years. Tracking could be applied to various tasks, for example motion-based recognition, which aims to identify humans, or objects, in different scenes, with automated surveillance that aims to detect abnormal behaviour in the alert area, in the case of a security problem and vehicle navigation, traffic monitoring, etc [17]. There are always difficulties in the tracking task, due to the noise of an image, complicated movement of an object, object occlusions or sudden illumination changes, as well as the computational cost of processing.

Many methods have been proposed to solve the tracking problems in various scenarios. To track an object, the first thing that needs to be done is to choose a way to represent the object. Points are the most commonly used representation method for objects. It could be represented by the centroid, or by a set of points [18], or could be represented by Primitive geometric shapes [19], silhouette and contour [20] [21] [22], articulated shape models [17] and skeletal models [23]. As well as choosing the representation methods, it is also important to decide which features will be used in the tracking systems. Colour feature has been widely used in most works [19] [24] [25] and other features, such as edges, optical flow and textures are also used in different works [17].

In the study led by Ismail Haritaoglu and David Harwood [21], in order to reduce the computational cost, they choose to use greyscale images, instead of colour images, and combine shape and some robust techniques to track and locate people. Two models have been applied in their tracking algorithms. The first is used to keep tracking the objects that show up in the foreground area, while the second is used to estimate the object location in the series of frames. They also choose a median coordinate, rather than a centroid, to represent the objects, as this is more robust to large motions. In

another study proposed by Jianpeng Zhou and Jack Hoang [26], they choose to use colour, aspect ratio, edge and velocity to represent humans and then use codebook to model. Kalman filter has also been used in their work to make the predictions of tracking.

It is possible to take the tracking task as a learning procedure and to implement the machine learning technique. The BP network was trained to recognize a pedestrian in [27], and CNN, using both spatial and temporal features, was used to track humans in [28].

2.4 Classification technique

Classification is a process that specifies the probability distribution to an observation that is given a class label. The most likely class with a high probability will be assigned to that observation [29]. Three well-known probability models are both introduced and discussed in this section.

2.4.1 Naïve Bayes Model

The Naïve Bayes Classifier is a simple machine learning approach for classifying observations to different classes and which assumes that variables (observations) are independently given the class label [29]. The probability of Naïve Bayes is a conditional probability computed as follows:

$$\mathrm{p(y = class|x)} = \frac{p(y)p(x|y)}{p(x)} \tag{2.17}$$

x and y here indicate observations and class, respectively, while p(x) is actually a normalization constant, which can be ignored. Then, the formulation could be rewritten as:

$$\mathrm{p(y|x)} = p(y)p(x|y) = \mathrm{p(y,x)} \tag{2.18}$$

The product rule is applied here to simplify the result. However, it is still not straightforward to compute the probability. By using sum rules and making the Naïve assumption, which assumes that the observations are all independently given the class label, the classifier could then be written as:

$$\mathrm{p}(\mathrm{y}|\mathrm{x}) \propto \mathrm{p}(\mathrm{y},\mathrm{x}) = \mathrm{p}(\mathrm{y}) \prod p(x_i|y) \tag{2.19}$$

By using this Naïve Bayes Model, the probability of the position and other features belonging to each state can be calculated and this probability could be used to decide the label of that image.

2.4.2 Hidden Markov Model

The Hidden Markov Model has been popular in recent years for labelling sequence structured data, especially in part-of-speech tagging [30]. In the Naïve Bayes approach, only the current image is considered, whereas, in the HMM model, the transition probability is also taken into account. An HMM could be thought of as a sequential extension to the Naïve Bayes Model [29]. Its equation could be formulated as:

$$\mathrm{p}(\mathrm{x},\mathrm{y}) = \prod_{i=0}^{n} p(y_i|y_{i-1}) p(x_i|y_i) \tag{2.20}$$

while $\mathrm{p}(y_i|y_{i-1})$ is the transition probability and $\mathrm{p}(x_i|y_i)$ is the emission probability. The transition probability computes the probability of the current state given the previous, while the emission probability computes given the states how likely it outputs the observation. With this model, it is capable to use context between images to make predictions which in turn may improve the accuracy of labelling.

2.4.3 Conditional random field

Different from the Naïve Bayes or the HMM, the conditional random field is the current state-of-art technique which considers the dependency of variables. It has been successfully implemented in many applications such as text processing, bioinformatics and computer vision [31]. Considering that this method is modelling the condition probability but not a joint probability as Naïve Bayes or HMM, the result of implementing this might have better accuracy.

2.5 Data set

The video data is a sequence of image taken from a stationary camera of ten days with framerate 1 frame/sec and the resolution is 1280 x 720. The images are 8bits depth RGB colour images with distortion caused by fish-eye effects. Total number of frames

are 205598 with the average frames approximate 20000 frames for each day.61754 frames of the date have no one inside the office while 55337 and 88507 frames for more than two persons and only one person inside the office res. Pre-process is needed to acquire useful information.

Chapter 3

Data acquisition

The data was taken from a fixed camera with resolution 1280 x 720 over 10 days in the same office area. The total number of frames is over 200000, with an average of 20000 frames per day. What are required from the data are the positions of the person and his behaviour. Due to the large amount of data, it is impossible to acquire requisite information manually. Therefore, algorithms for the acquisition of data are essential.

Two algorithms are illustrated below, with one for labelling positions and the other for labelling behaviours.

3.1 Positions labelling

Clicking through all the data to get the positions of the person is the simplest way to fulfil the task. However, even under the tolerance of missed labels and assuming that people could label 3 frames per second, it still takes more than 18 hours to complete the work. In order to work effectively and to spend a minimum amount of time in this task, a programme that can automatically label the images is used with the total labelling time to around 6 hours. The basic idea of the algorithm is to compare image difference. If the difference between the two consecutive images, or between separated images, is less than a threshold, it can be concluded that the person in the frame does not move, or that the movement is acceptable. The programme will then label the current processing frame to be in the same position as the image it compares to.

The algorithm first initiates the variable for saving the position and frame information, followed by comparing the current frame with the one with eight frames after that

frame. A normalize procedure will be applied to both frames to compute the absolute difference. When the difference is over a certain threshold, it is said that some changes have been detected through this period and, hence, manually labelling the positions for the frame is necessary. If the difference is less than a threshold, the programme will label the positions in the current frame the same as the one with eight frames before it. The detailed flow chart of the programme is shown in Figure 3.1

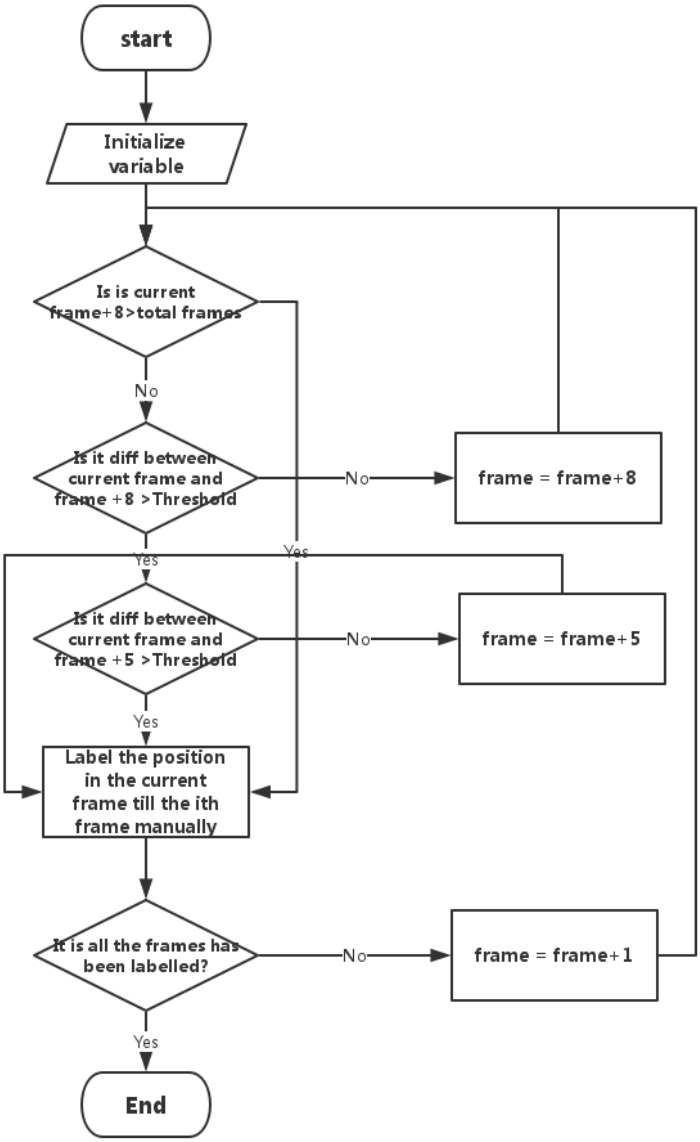


Figure 3.1: Flow chart of positions labelling

The problem with this labelling method is that, when the person in the frame quickly move to another position and then goes back to his original position, the frame between these actions might be taken as not moving or, even the difference is less than the threshold, the person does move (this happens when only a small part of the human body is detected). It is also possible that a frame, or a few frames, may skip the labelling, as the program require the use of a mouse to click to get the positions information then the use of a keyboard to jump to the next frame. It could be the case that, sometimes, the keyboard will be pressed first before clicking the mouse, which would cause an error in the labelling.

3.2 Behaviour labelling

There are 12 different states of behaviours referring to: ‘No one’, ’Entering’, ‘Leaving’, ‘walking’, ‘using terminal’, ‘sitting’, ’near whiteboard’, ‘near windows’, ‘near bookshelf’, ‘talking to someone’, ‘abnormal behaviour’ and ’Not define’. Similar methods to the positions labelling were used in this task, however, in considering that the positions information was collected before it was possible to use the positions information to label the behaviour, instead of comparing the image difference, here the algorithm simply compares the positions between the two frames. In this way, labelling through all the frames takes approximately 3 to 4 hours.

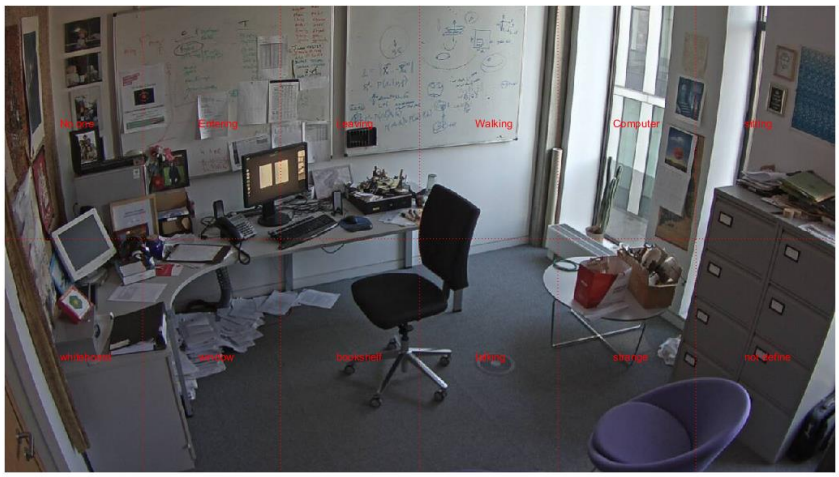


Figure 3.2: The image is separated into 12 regions, by clicking on the specify image a label will be give the that frame

All the frames are labelled as ‘No one’ at first and, by running the programme, it computes the Euclidean distance of the positions in the current frame and the one with eight frames after that frame. The threshold is set to be 5 pixels. A distance of less than 5 pixels indicates that no changes have been observed and the programme will then label the current frame to have the same label as the previous frame. The image has been separated into 12 regions and clicking in the specific region will give that frame the same label as is shown on the image. More labelling errors might occur, as this programme relies heavily on the positions information. An error in the positions information might cause errors in this data. Hence, the ground truth correction methods would be introduced.

3.3 Data correction

Due to some unexpected incidents, the ground truth will contain errors, including both labelling with wrong information and omitting to label. Errors in the ground truth will cause the estimation of the following programme to be inaccurate and, hence, a method is used to solve this problem.

Firstly, the behaviour data has been assumed to be correct before the positions, for the reason that even though the behaviour labelling is heavily reliant on the positions, it will not miss any subtle changes in the positions and, also, if the behaviour labelling is skipped in the labelling process, it will be automatically be labelled as ‘No one’. To correct the behaviour label, it is necessary to determine which label is wrong. There are two conditions in which an error may occur. One is when the positions label indicates that the person is at the (0, 0) point, which is the initial value of the positions, but the behaviour label is not ’No one’. Another condition is when the behaviour label is ‘No one’, but the positions label contains a value other than (0, 0). By listing these two errors and re-labelling them, it is reasonable to believe that the accuracy of the ground truth of behaviour will improve.

Secondly, the behaviour data is used after correction to re-label the positions in which the behaviour label is ‘No one’, but the positions label is not (0,0). In considering mistakes that may be missed by the positions process, a programme to smooth the positions is used. The algorithm detects if the positions label of the person disappeared

in a frame or two and then re-appeared within 2 frames. If that does happen, the algorithm would label these one or two frames to be the same as the previous one. The ground truth of the number of frames of the behaviours in each day is shown in Table 3.1

	1	2	3	4	5	6	7	8	9	10	11
Day1	7314	39	40	192	13064	553	0	0	0	6033	0
Day2	3189	15	26	445	2751	85	0	0	0	6571	0
Day3	6263	11	25	173	7313	1640	0	0	2	5104	0
Day4	3837	3	2	93	3900	332	0	0	3	4451	31
Day5	4490	17	17	1419	9912	1590	0	0	14	11310	0
Day6	11098	19	39	563	4665	2407	0	5	10	9341	10
Day7	5611	11	8	192	7417	962	0	6	22	2012	9
Day8	4273	6	8	120	4148	1463	0	0	0	6212	23
Day9	8669	23	37	294	9523	1295	0	7	3	175	299
Day10	7010	26	47	360	7434	3315	0	2	10	4128	12

Table 3.1 Numbers of frames of each behaviour in each days

Chapter 4

Methodology

The human behaviour labelling task has been separated into five submissions: background subtraction, which illustrates the algorithms for extracting the foreground from a noisy background; tracking, which illustrates the features and methods used in tracking; background update, which demonstrates the background update rules; and feature extraction and classification that is used for classifying human behaviour, given the feature.

4.1 Background subtraction

Different algorithms of the background subtraction algorithm have been introduced in the background chapter. A background subtraction task could be separated into 3 parts, including algorithms for sudden light change, algorithms for subtracting the foreground and algorithms for cleaning noise in the foreground image.

The concept in [21] [22] uses a greyscale image to subtract the foreground, however, this is not applicable in this project, as, although it works well with gradual light change or it could work in a sudden change, a delay in correct subtraction would occur. Figure 4.1 shows the subtraction using the algorithm from [21], under constant or gradual light change.

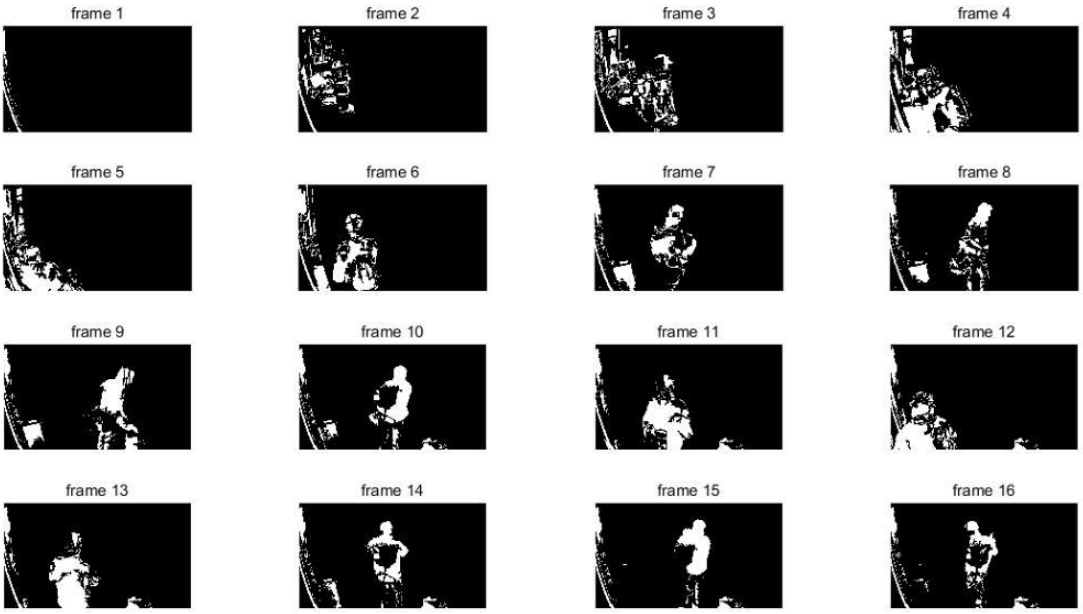


Figure 4.1: Subtraction under constant light.

It can be seen from the result that this works well in constant light conditions, even if there are some shadows affecting the subtraction, but, when compared to the conditions with sudden light change, it does not perform well enough. See Figure 4.2 from frame 1142 that shows when the light in the room in turned on the subtraction algorithm works poorly.

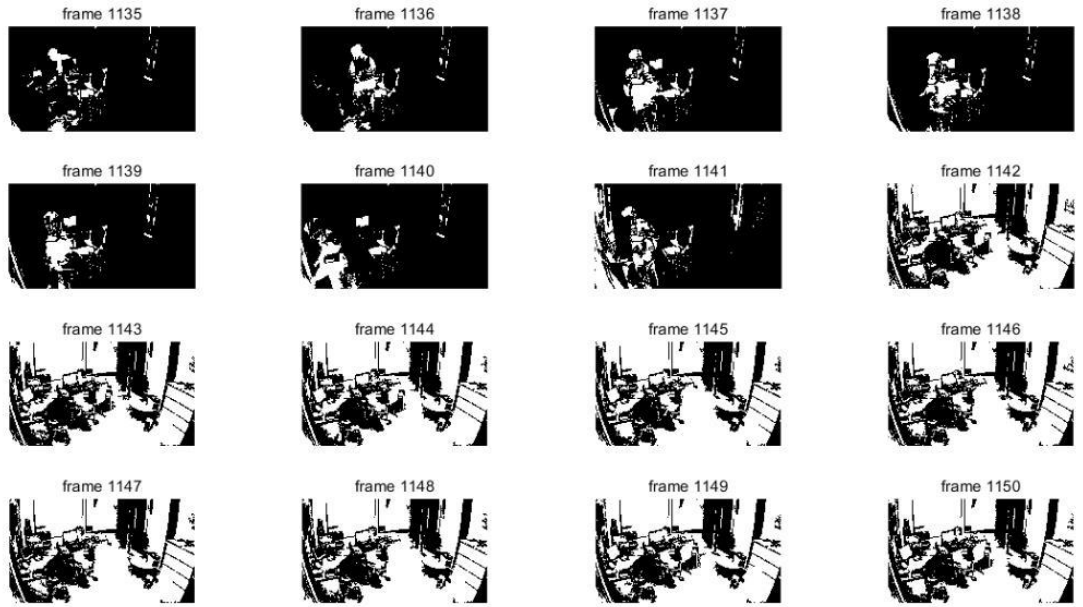


Figure 4.2: background subtraction under sudden light change

By using Gaussian mixture model to build adaptive background model, it could work satisfying satisfactorily under a constant, or gradual, light change. See Figure 4.3.

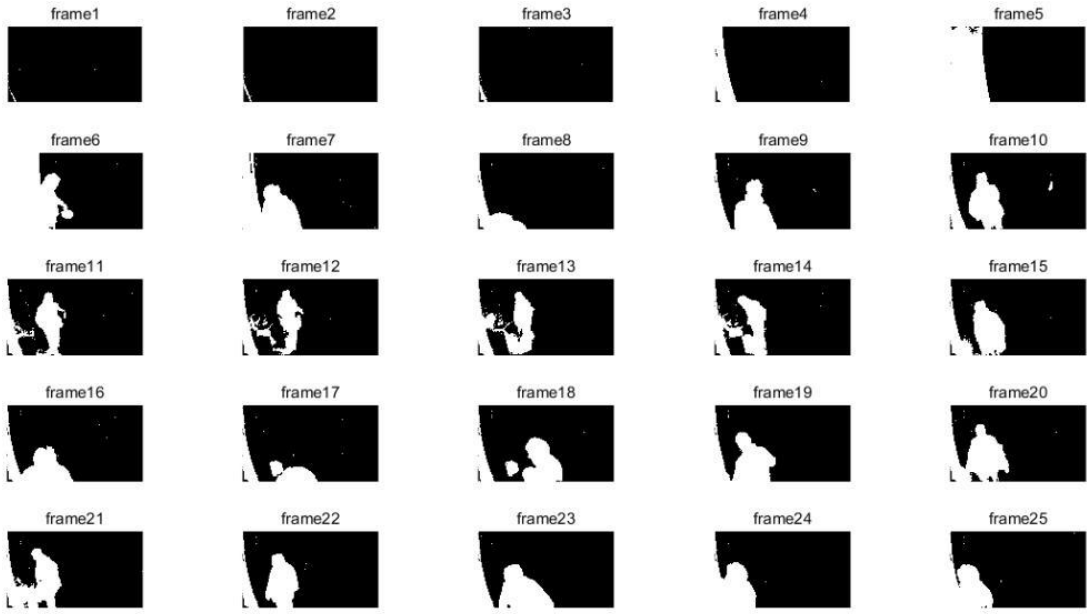


Figure 4.3: GMM works under constant light condition

However, if the lighting condition changes, the background model will not be able to adapt instantly. See Figure 4.4.

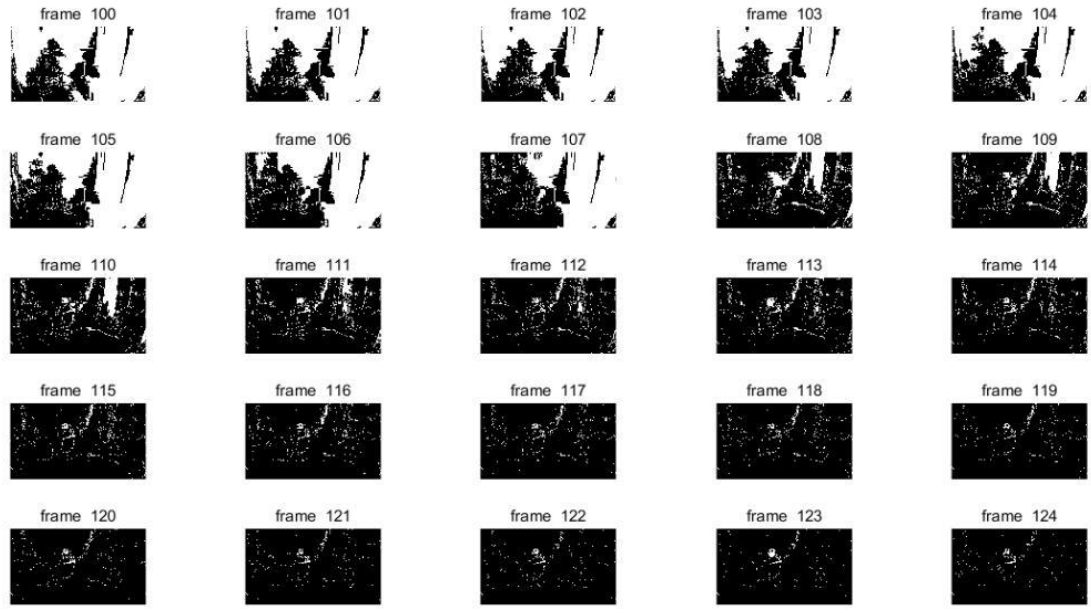


Figure 4.4: GMM with sudden light change and adapt to it

After certain frames the background model will be capable of fitting to that scene. The dissatisfactory element is that the person inside the scene will also be updated to the background model, no longer being detected by the system. See Figure 4.5

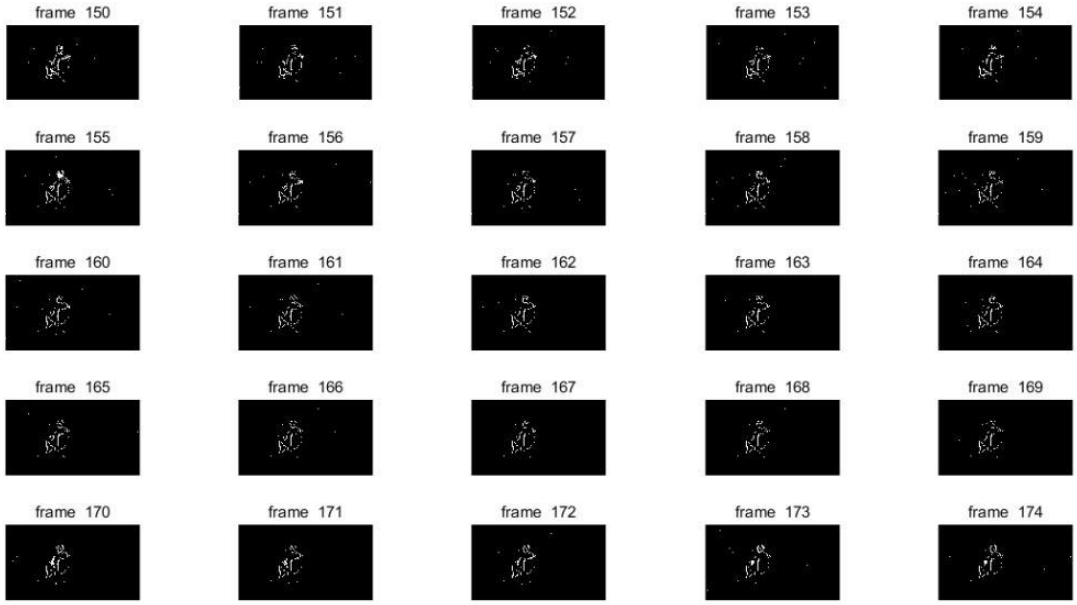


Figure 4.5: GMM, the person in the scene becomes the background

The above method in background subtraction cannot meet the requirement, hence, this project proposed another algorithm. To reduce the effect of illumination changes, in this project the normalized RGB colour space is used. The equation of normalizing selects a simple version, as follows:

$$\begin{cases} \mathbf{I_R} = I_R/(I_R + I_G + I_B) \\ \mathbf{I_G} = I_G/(I_R + I_G + I_B) \\ \mathbf{I_B} = I_B/(I_R + I_G + I_B) \end{cases} \tag{4.1}$$

The image difference is the absolute difference between the current frame and the background frames. The background model is updated after a certain period and this will be discussed in section 4.3. The noisy foreground is then obtained by determining whether the image difference is larger than a pre-determined threshold. Morphology operations, such as open, close, erode and dilate, are then applied to the noisy background to remove the noise and connect the component. Due to the reason that the image frame taken by the camera has a fish-eye effect, the lines are all blended, and using prior knowledge of the layout of the room to remove such objects as the

door becomes difficult. Therefore, a colour based procedure is included to remove the door from the background instead, in order to separate the person and the door when the person is entering the office. The colour of the door is obtained by a Gaussian model trained with 50 frames of the door colour. The result of subtraction is in Figure 4.6

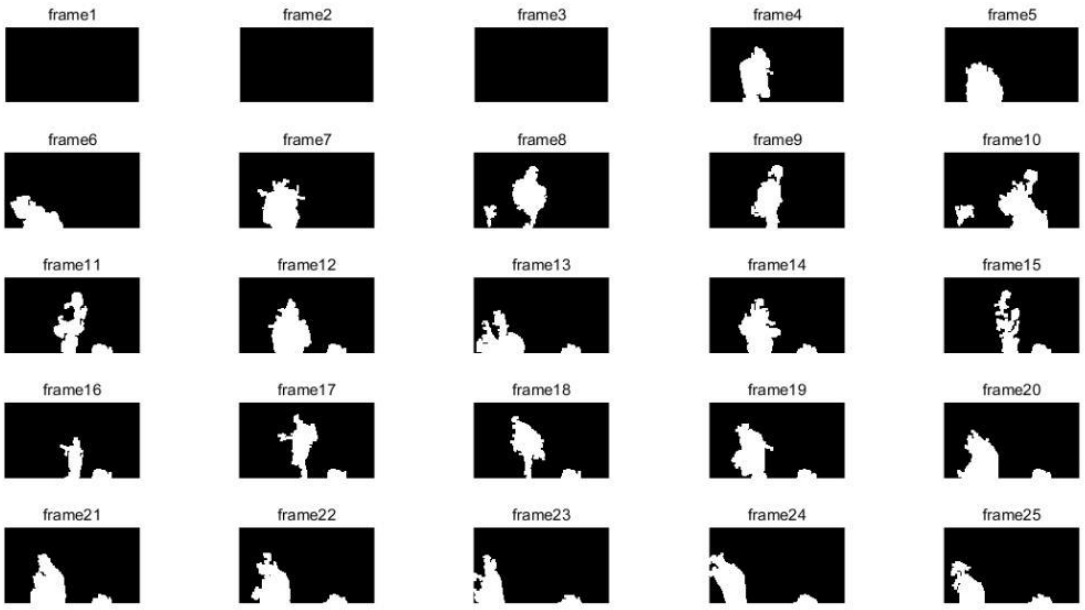


Figure 4.6: Background subtraction result

The figure of the person in the image is clearly seen and less noise can be observed. Some small objects will show up in the image as the background does not update, but this method could work under sudden illumination change in a certain range. Even though the person's body does not always appear as complete, it is enough for extracting useful information. After the tracking part is introduced, the foreground model may more stable and small objects can be removed.

4.2 Tracking

The tracking algorithm initializes when the person enters the room and a label will be given to that person. The feature used for the tracking process is only the colour histogram. Euclidean distance is not appropriate in this case, as the frame rate is 1 frame per second and a frame or two will sometimes be missing, which causes the person to suddenly appear somewhere. Rather than using the Euclidean distance, in

this project the Bhattacharyya distance of colour histogram is calculated. By finding the most possible object coming from the previous frame to the current frame, it is capable of constantly tracking that object. Failed tracking will occur in some cases when the background subtraction does not work properly and, if the person in the frame separates into two parts, then only the part with higher probability will be considered as being the person. Lost tracking happens after the case above occurs, as, if part of the person's colour is the same as the colour of objects in the room, the program will take other objects in the room to be the person, until the person interacts with these objects again. Figure 4.7 shows the 12 frames of the tracking process. The bag, which is not a background object, does not get tracked during the process.

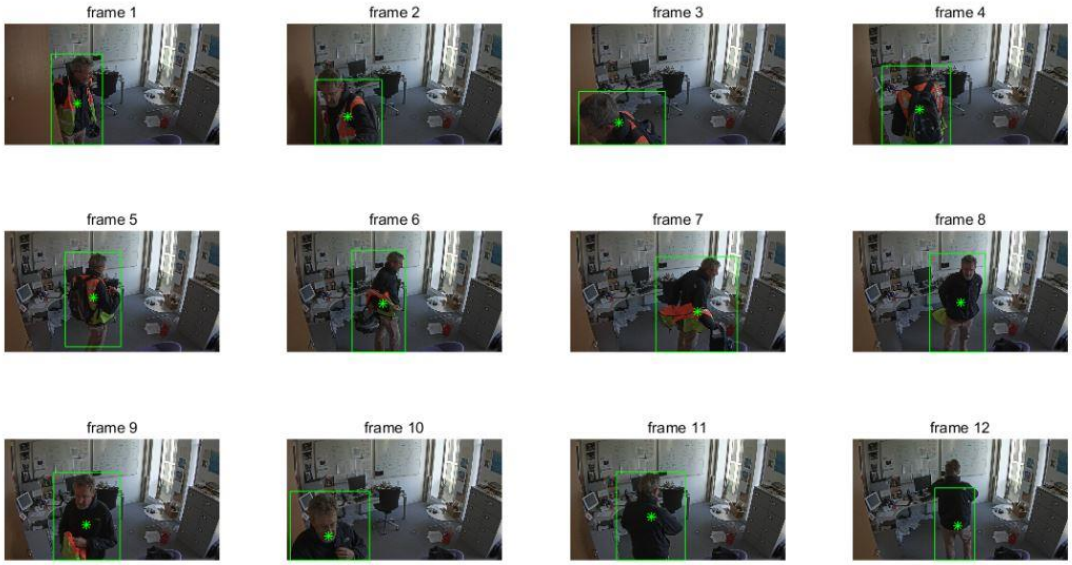


Figure 4.7: Tracking with bounding box and centroid

The computational cost in obtaining the histogram of the target is very large, sometimes taking over 10 seconds. In order to accelerate the algorithm, instead of using the whole target area for tracking, it uses only a small rectangle area to share the same centre as the target, but the area is only '0.3 x MinorAxisLength x MinorAxisLength'. The 'MinorAxisLength' is calculated by the Matlab built in function 'regionprops', which is the minor axis of the ellipse that could cover the region area. In this way, most of the time, the computational cost for each frame can be controlled within 2 seconds.

4.3 Background update

The algorithm that is programmed for updating the background combines the background subtraction methods and the tracking methods. The background model is firstly chosen as the first image of the scene. When a person is detected, which means the foreground has been extracted by the subtraction algorithms, the tracking algorithm will soon indicate which figure is the person and a list of the pixels representing the person will be obtained. The list will then be used to remove the objects beside the figure of the person. The colour histogram will be calculated again here to store the colour information in the current frame and will be used to track the person in the next frame. After a period of time, the background model will automatically update the background outside the bounding box of the figure of the person to be the current frame.

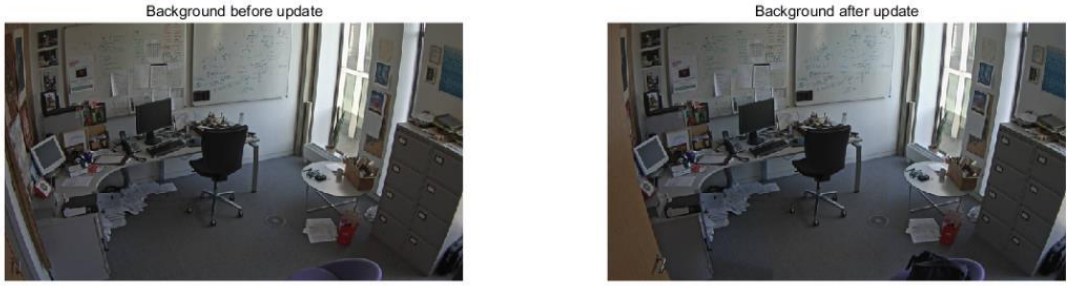


Figure 4.8: Background before and after update

It can be seen that, after running the background update algorithms, the door is opened and a bag has been left on the sofa.

4.4 Feature array

The feature for representing the person in the room is the centre of the person, the speed between consecutive frame of the person and a value that indicates how many people are in the room. In this project, all this data is acquired by using the ground truth data as the detection algorithms illustrated above is not robust enough and the computational cost is still too large to implement. The defect and evaluation will be analysed in the evaluation chapter. After the feature has been extracted, it will be concatenated into arrays in the order of: vertical positions (Positions_x), horizontal positions (Positions_y), speed between the positions of two frames (Speed), speed on vertical axis between two frames (Speed_x), speed on horizontal axis between frames (Speed_y), value indicates the number of people (0 for none, 1 for 1 people, 2 for more than one people) (Indicator). A typical feature array Shown in Table 4.1:

Feature	Positions_x	Positions_y	Speed	Speed_x	Speed_y	Indicator
Value	552.75	367.25	9.124	1.5	8.999	1

Table 4.1: Typical feature array

The histograms of position and speed features plot under each behaviour are shown in Appendix A.

4.5 Behaviour Classification

The hidden Markov Model is implemented in the classification task rather than selecting the simple model of Naïve Bayes. The HMM algorithm is from the built in toolbox of Matlab. The toolbox is literally for discrete values, considering the value in this project has an upper limit as well as the value besides speed should be integers it is reasonable to use this toolbox. All the feature values are turned into integers before implementation. Since the largest speed value should not exceed the largest distance in the scene, which is the diagonal of the image that is 1468, the speed value is assigned to 735 bins. 10 folds cross-validation is implement in the training process. The 10 day data is separated into 9 training data and 1 testing data each circulation. The transition matrix and emission matrix is obtained with the ‘hmmestimate’ function from the toolbox by providing the feature and the corresponding label. 6 models are then trained for each of the features by using the ‘hmmdecode’ function which input value is the test sequence, transition matrix and emission matrix. Each model of the six is used to make predictions by combine all the posterior probability computes by ‘hmmdecode’ to find the largest value of each state. The state with highest probability is chosen to be the label for the corresponding image. The accuracy without giving any tolerance using ground truth for classification is shown below in Table 4.2:

Times	accuracy	Frames
1	0. 9621	27235
2	0. 9028	13082
3	0. 8902	20531
4	0. 9405	12652
5	0. 8917	28796
6	0. 8844	28157
7	0. 9274	16250
8	0. 8971	16253
9	0. 6992	20325
10	0. 8324	22344

Table 4.2: result of HMM

The average accuracy of all the training is 88.28%. More experiment will be discussed in the next chapter.

Chapter 5

Experiments

In this chapter, experiments and an evaluation of background subtraction, tracking, and background update, as well as classification are discussed.

5.1 Evaluation of background subtraction

The background subtraction algorithm has been illustrated in the methodology chapter**.** In this section, some tests and an evaluation of the background subtraction will be provided.

Figure 5.1 shows 30 image frame from one of the days when the person enters and is walking in, the room.

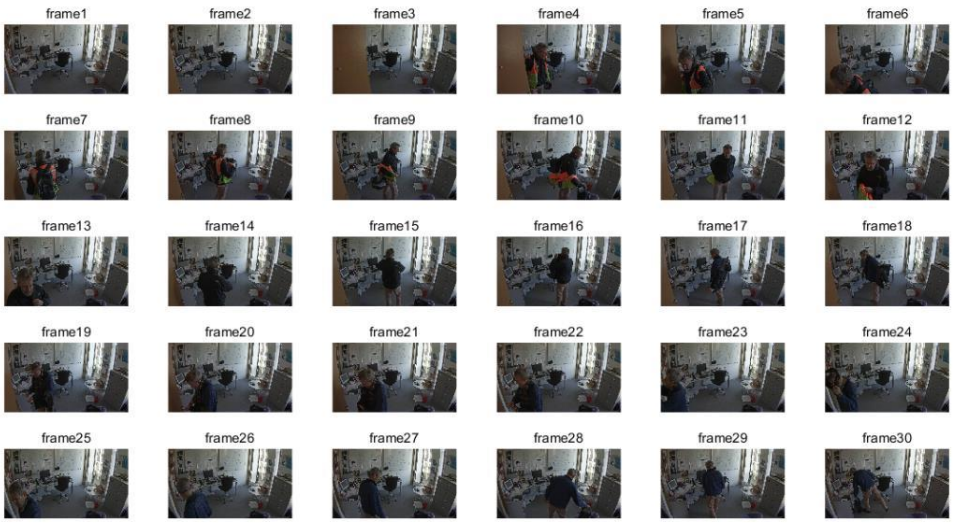


Figure 5.1: Video frames

It can be seen from the frames that the person enters wearing visible clothing and hangs it on the door, and a bag which is not a background object, is introduced into the scene. The foreground image, after implementing the background subtraction is shown in Figure 5.2. The algorithm works to provide an acceptable result with the new object and the person has been separated in most cases. When the person is close to an object that does not belong to the background, the algorithms will not be able to separate them. Also, as the threshold is pre-determined, some parts of the foreground will sometimes be in the background, and especially in cases where there is a black or white object, whether it is a new object or a background object. If the black or white object is a new object it might disappear as in normalized RBG black, white and grey are all have small value, or if it is a background object, the black or white object in front of it will not be detected, or will split the foreground (see frame 22 in Figure 5.2).

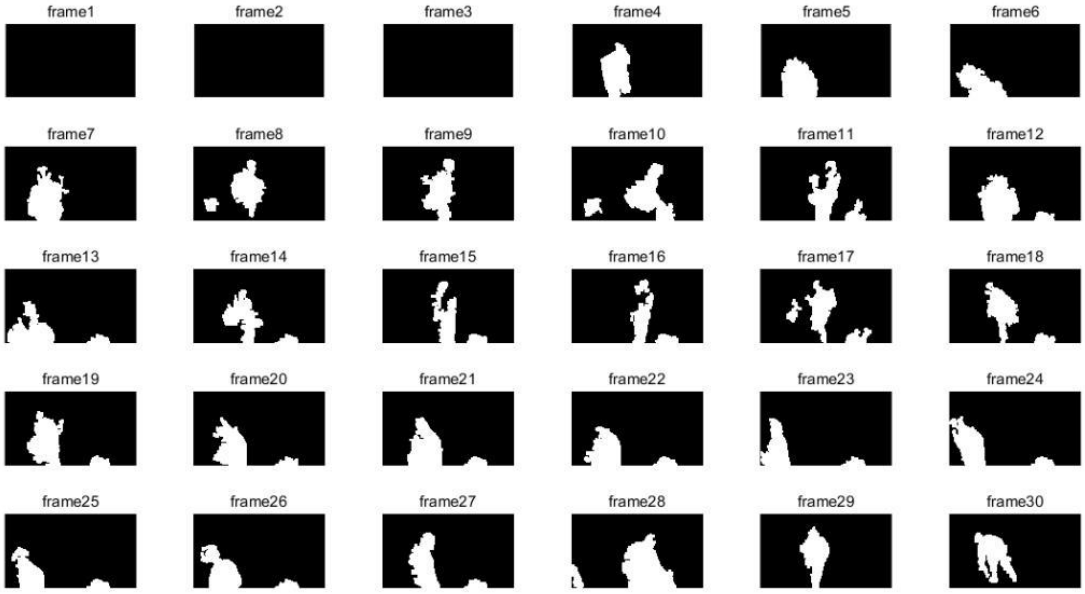


Figure 5.2 Background subtraction under gradual light change

Unfortunately, it is not always the case that the background subtraction could works well. The algorithm is capable of handling constant light, or gradually changing light conditions, and, in a reasonable range it is invariant to the light changes. However, it does not have the capability of dealing with shadows or reflection. Figure 5.3 demonstrate a case where the light in the room is turned on when the room is dark.

Figure 5.3 Sudden light change with background subtraction

The result in this case is terrible. The region of the person grows and connects to other areas, and many objects that do not belong to the foreground appear. Being caused by reflection and shadows, the window, cabinet and even the white paper on the desk and ground appear in the foreground, although this still works better than GMM and MaxMin methods illustrated in the previous chapter. Algorithms that could supress the effect of shadows and reflection are needed desperately.

5.2 Evaluation of tracking

A comparison of tracking accuracy of 1000 frames with or without background update is given in Table 5.1. Accuracy is estimated by the distance between the centre from tracking and ground truth. Distance less than certain pixels between tracking centre and ground truth is accepted as hit. With 2% tolerance, the accuracy in tracking without background update is only 25% whereas is over 90% with background update. The average in tracking with background update always outperforms the pure tracking algorithm.

Tolerance	Accuracy without update	Accuracy with update
2% (30 pixels)	25%	91.55%
5% (73 pixels)	95.82%	96.33%
8% (117 pixels)	99.49%	99.66%
10% (146 pixels)	100%	100%

Table 5.1 accuracy on tracking

In the pure tracking algorithm, objects with similar colour histogram with the person might result in a miss tracking from time to time. lightning condition might also affect the colour histogram which cause unexpected tracking. However, when background

update is added, small objects will be estimated as background and will not appear in the foreground to confuse the tracking algorithms which in turn improves the accuracy of tracking.

Background update rate is a crucial factor in deciding the accuracy. Smaller rate will have better result with background changing and react quicker to new objects or lightning change that will increase the accuracy but when the background subtraction method fails or the tracking fails foreground will go into the background which cannot be fixed during the whole process. On the contrary, larger rate will be more robust to error in background subtraction or tracking but insensitive to new objects which influence the accuracy of tracking.

5.3 Evaluation of classification

The accuracy of the classification by comparing the prediction label with ground truth label is given in Table 4.2 in chapter 4. The accuracy that been demonstrate before estimates the frame to frame label precision, however, the classifier using Hidden Markov Model sometimes label an image with correct label a few frames ahead or later. Therefore, the frame to frame estimation on the accuracy obtained by HMM is not precise. Considering the defect that a HMM has, it is sensible to have a tolerance for estimating the accuracy. The tolerance here means, if 1 frame tolerance is given, the prediction label is correct if a label a frame before or after that image is correct comparing to ground truth label.

Figure 5.4 shows that the accuracy improves with larger tolerance and it reaches 100% after seven frames tolerance. In most of the days the prediction without any tolerance is around 90%, only in day 9 and day 10 the accuracy is 70% and 83% respectively. There are two explanations, the first one is the behaviour changes too fast the HMM cannot give precise label in time, the second one might be the ground truth is noisy which means there are some error label in the ground truth data.

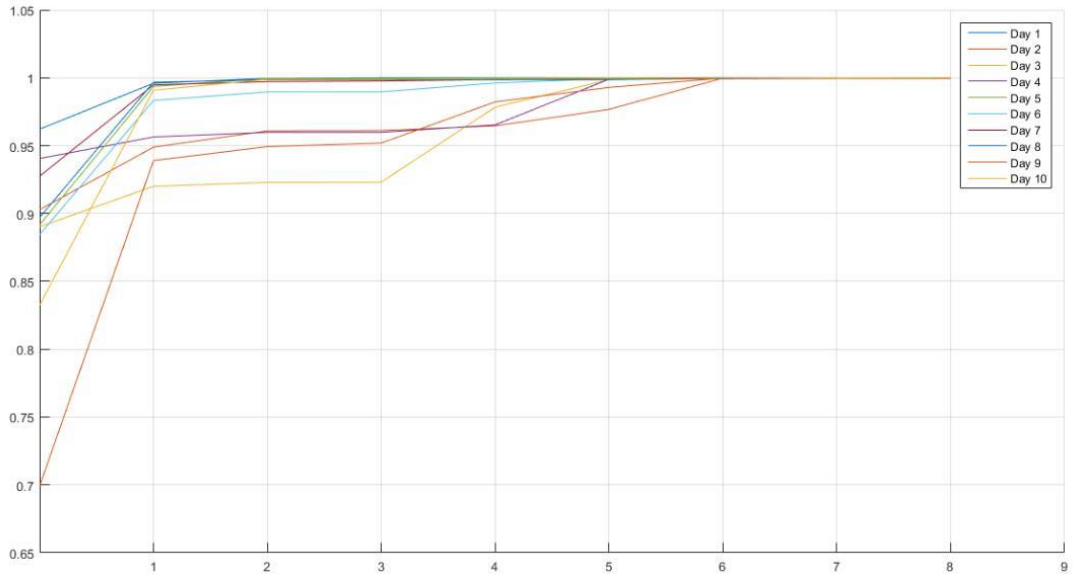


Figure 5.4 Accuracy with given certain tolerance (the vertical axis is the accuracy while the horizontal axis is the frame tolerance)

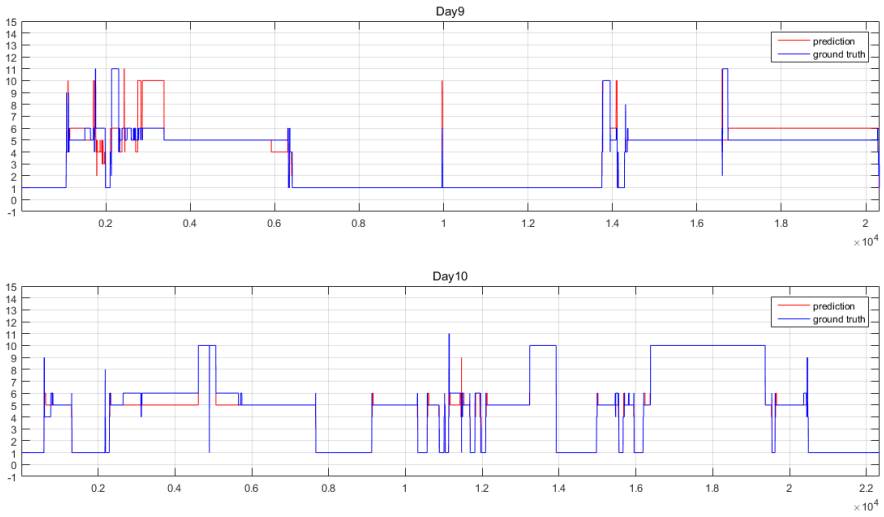


Figure 5.5 Prediction plot over ground truth on day 9 and day 10 (vertical axis is the behaviour label, 5 is ‘sitting’, 6 is ‘using terminal’, red line is from prediction, blue line is from ground truth)

It could be seen from Figure 5.5 that in day 9 and day 10 there are a short period that the person’s behaviour changes quite often (between 1000 to 3000 frames in day 9 and between 1000 to 1200 in day 10). Moreover, the classification cannot always

distinguish between ‘sitting’ and ‘using terminal’ behaviour which might due to the reason that features between those two are not enough.

To check whether the precision is caused by the inaccurate prediction from HMM or it is the error from ground truth 1000 frames which the prediction labels do not match the ground truth labels of day 9 have been re-estimated. In Table 5.1 it demonstrates the false positive and false negative as well as the wrong label in ground truth in 1000 frames which prediction does not match the ground truth. False negative here means the prediction is wrong while the ground truth is right and false positive means the prediction is right while the ground truth is wrong.

False negative	False positive	Both wrong
983	13	4

Table 5.2: Counts for classification labelling and ground truth behaviour comparing to real behaviours in 1000 frames

From Table 5.2 it could tell that the 98.3% of the label in 1000 when the classifier does not match the ground truth is wrong. Whereas, it is still noticeable that 13 frames of the HMM classifier produces the correct labels but the ground truth are wrong. And there are actually 17 frames out of 1000 are labelled with wrong behaviours (1.7%). By taking the account that there is wrong label in the ground truth the accuracy of the classifier could be re-estimated as:

$$\text{Accuracy} = \frac{N_t \cdot P_t + N_f \cdot P_f}{N} \tag{5.1}$$

Which N is the total number of frames, N_t is the total number of frames that the classifier is correct, N_f is the total number of frames that the classifier is wrong, P_t is the probability that the classifier is actually right while it the same of ground truth (2000 frames are checked carefully to see if there is miss label but all are correct $P_t = 1$), P_f is the probability that the classifier is actually right while it differs from ground truth ($P_f = 0.13$). The classifier accuracy computes by (5.1) increased to 89.8% from 88.28%. Figures that plot both ground truth data and classifier result has been shown in Appendix B.

Chapter 6

Conclusion

This project aims to classify human behaviour in an office environment by analysing video data. Ground truth data was collected by using the tools developed in this project. Several techniques including background subtraction, tracking, background update and a classifier relates to Hidden Markov Model was implemented in this project. The accuracy of labelling the image compared to ground truth is satisfying which could reach 88.28% and will be over 99% when tolerance over 6 frames is given.

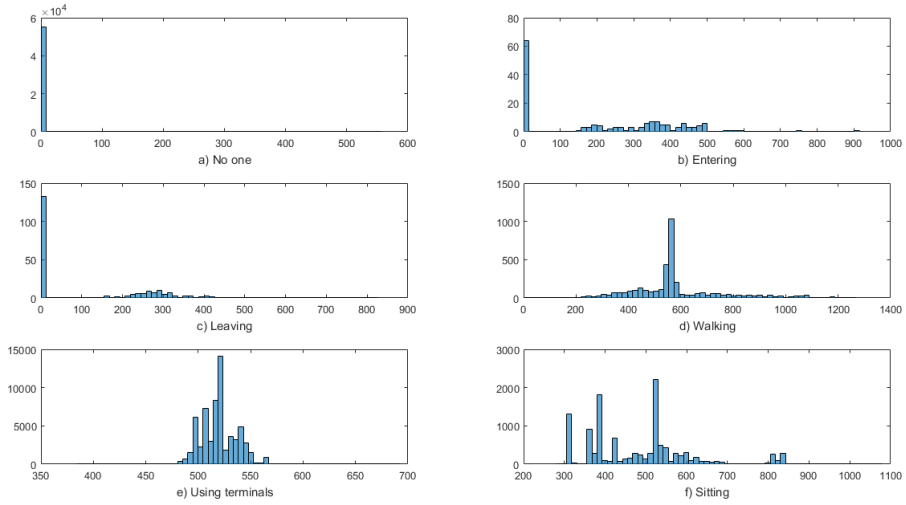
The background subtraction algorithm is based on RGB colour space. By using normalised RGB, it is more robust to lightning changes but still will suffer from large change in lightning conditions caused by reflection or shadows. Subtraction will fail to subtract foreground from background at that situation. It is suggested that the subtraction method instead of using colour should consider other foreground detection methods for instance using contours or kernel density estimation which will be more robust to complicated lightning conditions in real life data. Tracking algorithm works well in most cases while will fail when the lightning condition changes too much and the colour of the object in some situations shift. Only when the missed track object gets close to the target object will the tracking recover and track the target object again. Background update method relies heavily on the subtraction and tracking algorithms as it updates background after a certain period in the region with no tracking object. As long as the subtraction and tracking algorithms work properly the background update method performs well. Hence, the system for detection and tracking is not robust enough to handle real day data in some situation.

Classification algorithms is based on Matlab HMM toolbox. The selected features are position, speed and number of people (0 means no one, 1 means 1 people, 2 means more than one). Since the detection and tracking might corrupt from time to time in the project the classifier is trained by ground truth data and the result is acceptable with the accuracy of 88.28%. In the future study, the feature should be computed by the program itself and features should be reconsidered. In the classification task, behaviours between ‘sitting’ and ‘using terminal’, ‘sitting’ and ‘walking’ sometimes will be confused due to some similar features they share. For example, when people sitting at the desk near the terminal it is hard to distinguish those two behaviours. To improve the result from the classifier features from motion histogram or optical flow might help. Moreover, since Hidden Markov Model is a generative model which models joint probability using discriminative models such as conditional random field to model the conditional probability directly might as well improve the result on prediction.

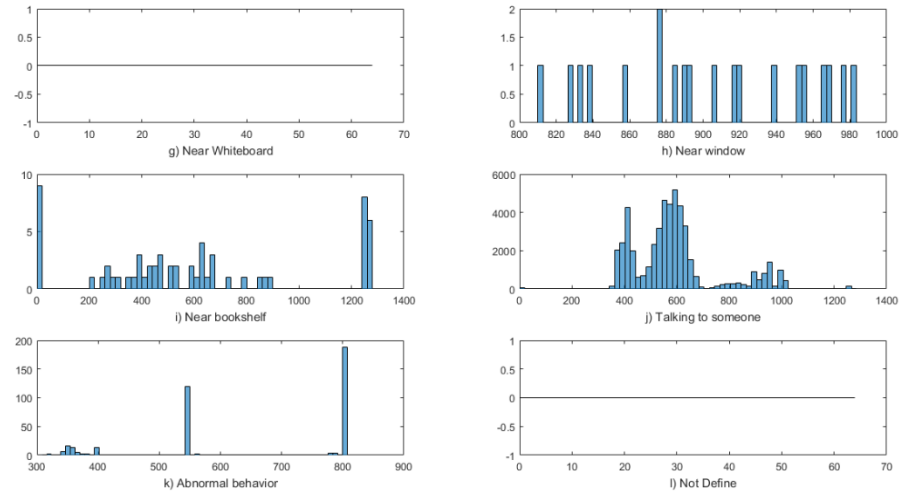
Appendix A

Histograms of features in each behaviours

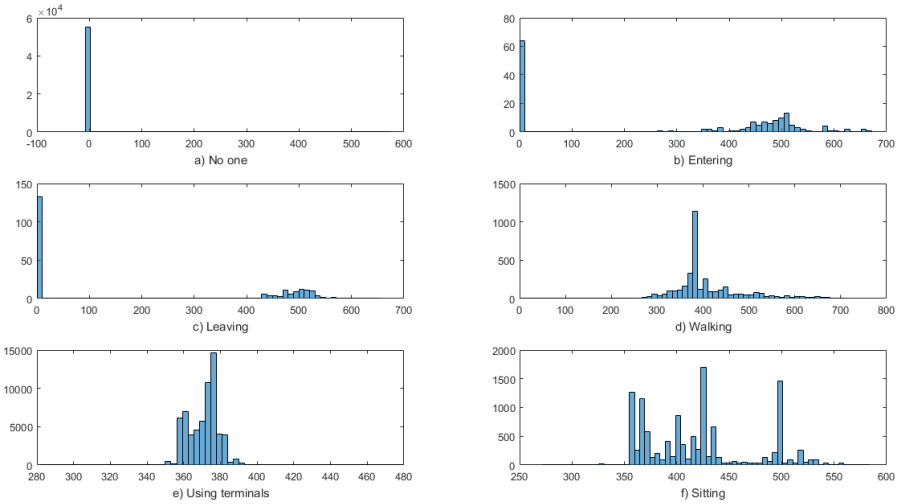
In this appendix, histograms of features including position and speed are plotted under 12 behaviours.



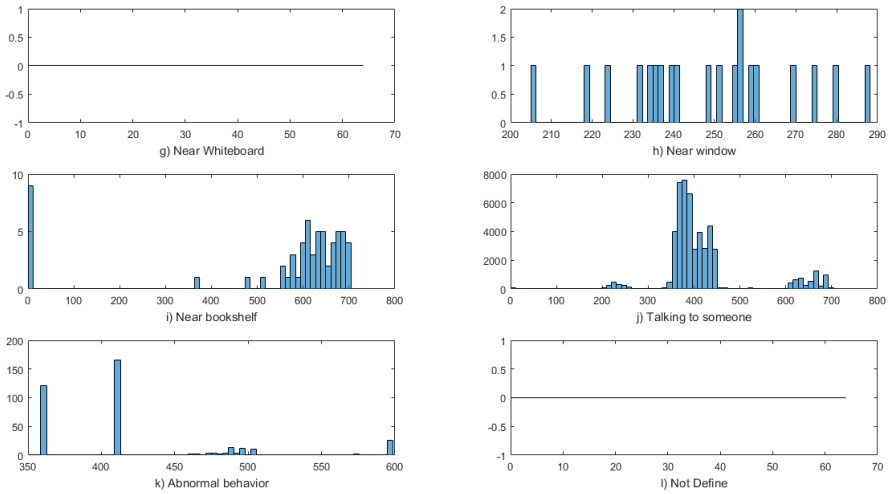
a) Histogram of positions of vertical axis



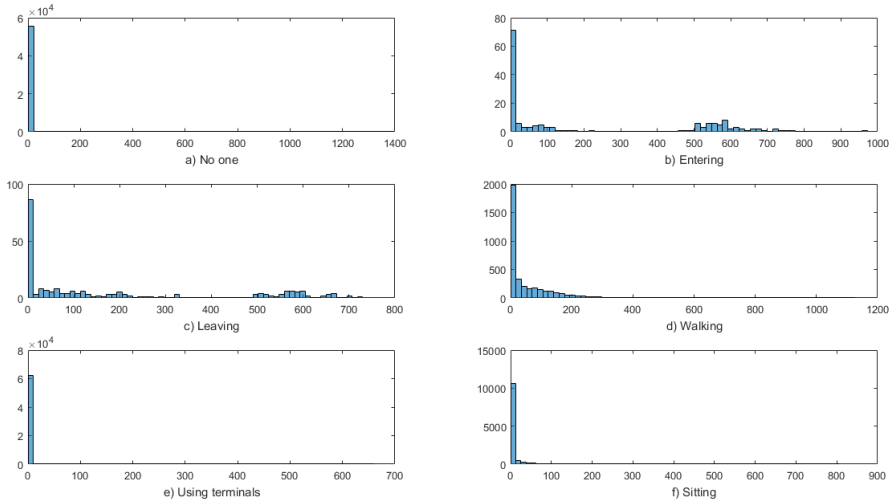
b) Histogram of positions of vertical axis



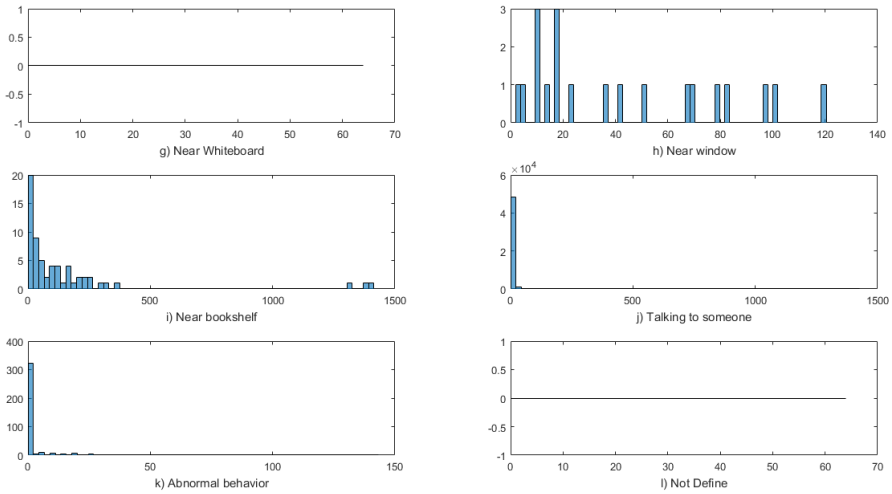
c) Histogram of positions of horizontal axis



d) Histogram of positions of horizontal axis



e) Histograms of speed

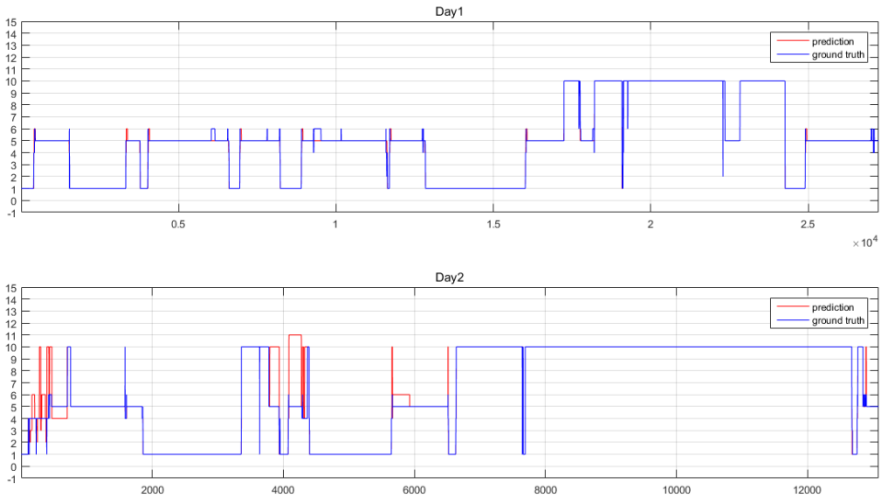


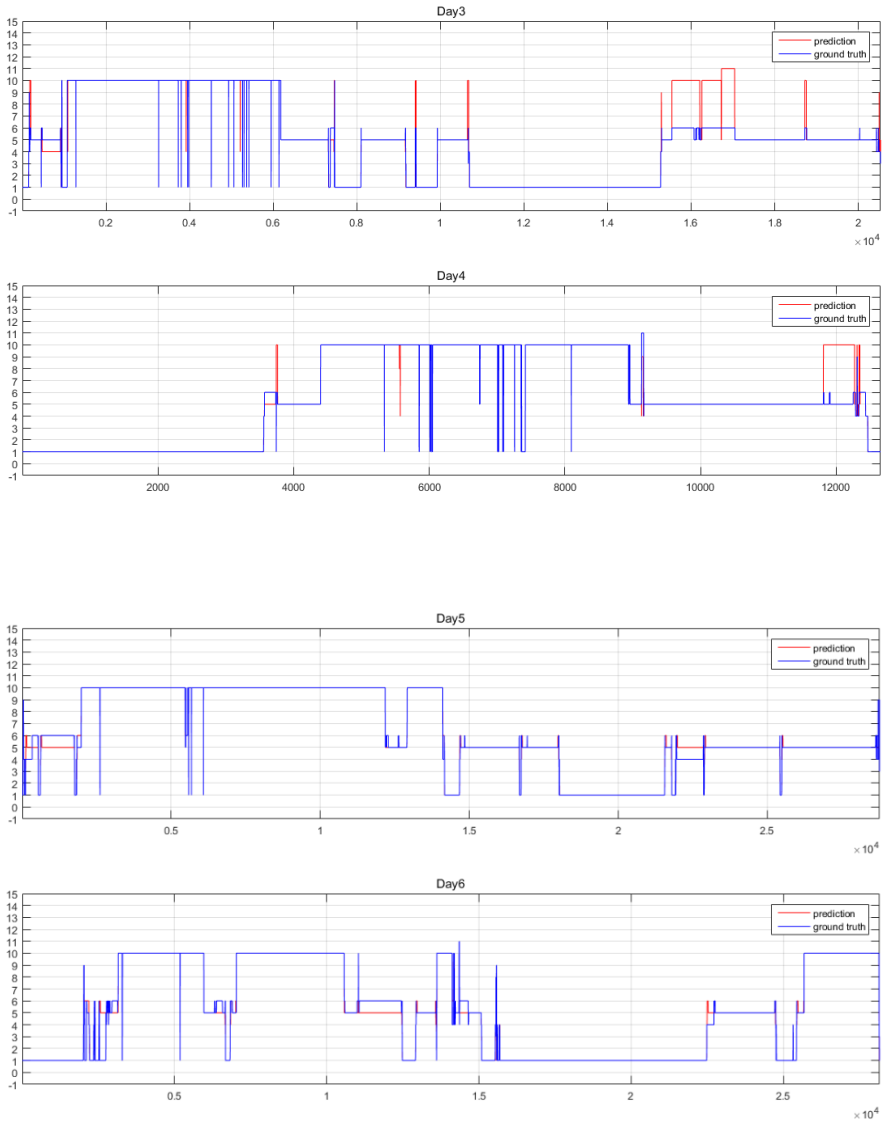
f) Histograms of speed

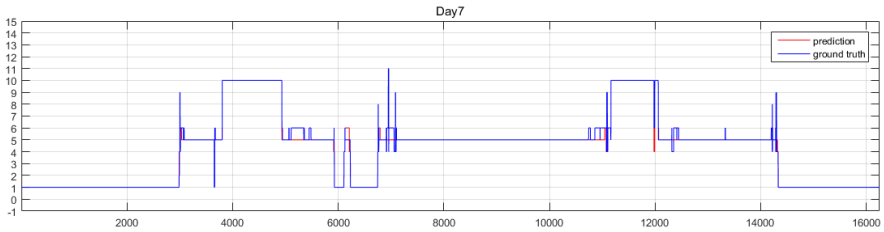
Appendix B

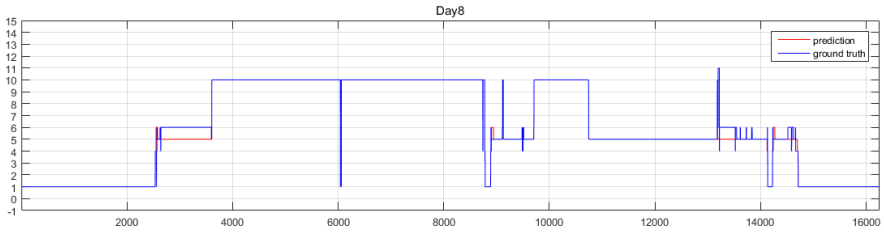
Classification results

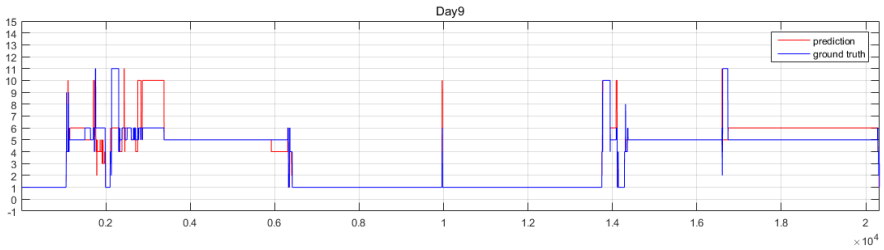
In this appendix, the classification results plot with its corresponding ground truth behaviours are shown.

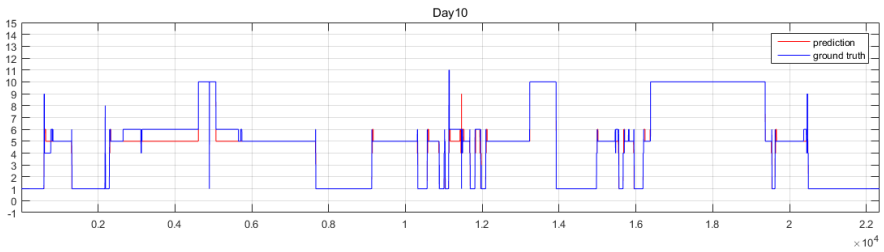



Day3
prediction
ground truth
Day4
prediction
ground truth
Day5
prediction
ground truth
Day6
prediction
ground truth


Day7
prediction
ground truth


Day8
prediction
ground truth


Day9
prediction
ground truth
×10^4


Day10
prediction
ground truth
×10^4

Appendix B. Classification results

Bibliography

[1] Huang, S.C., 2011. An advanced motion detection algorithm with video quality analysis for video surveillance systems. IEEE transactions on circuits and systems for video technology, 21(1), pp.1-14.

[2] Ayers, D. and Shah, M., 2001. Monitoring human behavior from video taken in an office environment. Image and Vision Computing, 19(12), pp.833-846.

[3] Kjeldsen, R. and Kender, J., 1996, October. Finding skin in color images. In Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on (pp. 312-317). IEEE.

[4] Fieguth, P. and Terzopoulos, D., 1997, June. Color-based tracking of heads and other mobile objects at video frame rates. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 21-27). IEEE.

[5] Yamato, J., Ohya, J. and Ishii, K., 1992, June. Recognizing human action in time-sequential images using hidden markov model. In Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on (pp. 379-385). IEEE.

[6] Umeda, M., 1982, October. Recognition of multi-font printed Chinese characters. In Proc. 6th ICPR (pp. 793-796).

[7] Huang, X.D., Ariki, Y. and Jack, M.A., 1990. Hidden Markov models for speech recognition (Vol. 2004). Edinburgh: Edinburgh university press.

[8] Tao J, Tan Y P. A probabilistic approach to incorporating domain knowledge for closed-room people monitoring[J]. Signal Processing: Image Communication, 2004, 19(10): 959-974. [9] Comparative study of background subtraction algorithms

[10] Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P., 1997. Pfinder: Real-time tracking of the human body. IEEE Transactions on pattern analysis and machine intelligence, 19(7), pp.780-785. [11] Adaptive background mixture models for real-time tracking

[12] Elgammal, A., Harwood, D. and Davis, L., 2000, June. Non-parametric model for background subtraction. In European conference on computer vision (pp. 751-767). Springer Berlin Heidelberg. [13] real-time foreground-background segmentation using codebook model

[14] Oliver, N.M., Rosario, B. and Pentland, A.P., 2000. A bayesian computer vision system for modeling human interactions. IEEE transactions on pattern analysis and machine intelligence, 22(8), pp.831-843.

[15] Han, B., Comaniciu, D. and Davis, L., 2004, January. Sequential kernel density approximation through mode propagation: applications to background modeling. In Asian Conference on Computer Vision (Vol. 39). [16] adaptive background estimation and foreground detection using kalman-filtering

[17] Yilmaz, A., Javed, O. and Shah, M., 2006. Object tracking: A survey. Acm computing surveys (CSUR), 38(4), p.13. [18] Resolving motion correspondence for densely moving points

[19] Comaniciu, D., Ramesh, V. and Meer, P., 2003. Kernel-based object tracking. IEEE Transactions on pattern analysis and machine intelligence, 25(5), pp.564-577.

[20] Yilmaz, A., Li, X. and Shah, M., 2004. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Transactions on pattern analysis and machine intelligence, 26(11), pp.1531-1536.

[21] Haritaoglu, I., Harwood, D. and Davis, L.S., 1998, June. W4S: A real-time system for detecting and tracking people in 2 1/2D. In European Conference on computer vision (pp. 877-892). Springer Berlin Heidelberg.

[22] Haritaoglu, I., Harwood, D. and Davis, L.S., 2000. W 4: Real-time surveillance of people and their activities. IEEE Transactions on pattern analysis and machine intelligence, 22(8), pp.809-830.

[23] Ali, A. and Aggarwal, J.K., 2001. Segmentation and recognition of continuous human activity. In Detection and recognition of events in video, 2001. Proceedings. IEEE Workshop on (pp. 28-35). IEEE.

[24] Wren, C.R., Azarbayejani, A., Darrell, T. and Pentland, A.P., 1997. Pfinder: Real-time tracking of the human body. IEEE Transactions on pattern analysis and machine intelligence, 19(7), pp.780-785.

[25] Sen-Ching, S.C. and Kamath, C., 2004, January. Robust techniques for background subtraction in urban traffic video. In Electronic Imaging 2004 (pp. 881-892). International Society for Optics and Photonics.

[26] Zhou, J. and Hoang, J., 2005, June. Real time robust human detection and tracking system. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops (pp. 149-149). IEEE.

[27] Zhao, L. and Thorpe, C.E., 2000. Stereo-and neural network-based pedestrian detection. IEEE Transactions on Intelligent Transportation Systems, 1(3), pp.148-154.

[28] Fan, J., Xu, W., Wu, Y. and Gong, Y., 2010. Human tracking using convolutional neural networks. IEEE Transactions on Neural Networks, 21(10), pp.1610-1623.

[29] Klinger R, Tomanek K. Classical probabilistic models and conditional random fields[M]. TU, Algorithm Engineering, 2007.

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

[31] Sutton C, McCallum A. An introduction to conditional random fields[J]. Machine Learning, 2011, 4(4): 267-373.

[32] Murphy, K. (2016). Hidden Markov Model (HMM) Toolbox for Matlab. [online] Cs.ubc.ca. Available at: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

[33] Avidan, S., 2004. Support vector tracking. IEEE transactions on pattern analysis and machine intelligence, 26(8), pp.1064-1072.

[34] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M. and Shafer, S., 2000. Multi-camera multi-person tracking for easyliving. In Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on (pp. 3-10). IEEE.

[35] Intille, S.S., Davis, J.W. and Bobick, A.F., 1997, June. Real-time closed-world tracking. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 697-703). IEEE.

[36] Collins, R.T., Lipton, A.J., Fujiyoshi, H. and Kanade, T., 2001. Algorithms for cooperative multisensor surveillance. Proceedings of the IEEE, 89(10), pp.1456-1477.

[37] Kato, H. and Billinghurst, M., 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on (pp. 85-94). IEEE.

[38] Ferrari, V., Tuytelaars, T. and Van Gool, L., 2001. Real-time affine region tracking and coplanar grouping. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 2, pp. II-226). IEEE.