# ABANDONED OBJECT DETECTION AND CLASSIFICATION

*By*


NC Samia Ahmed
NC Maemoona Kayani
NC Ayesha Kanwal
NC Waqar Ahmed


Submitted to the Faculty of Computer Software Department National University of Sciences and Technology, Islamabad in partial fulfillment for the requirements of a B.E. Degree in Computer Software Engineering
JULY 2011

# ABSTRACT

Video surveillance in public places has witnessed tremendous growth over the last decade with applications based on closed-circuit security systems to IP cameras. Among the various aspects of surveillance, automatic detection and classification of abandoned objects could serve as a valuable application reducing potential threats to public safety. The aim of this project is to detect the unattended abandoned objects from videos of public places and classify the detected objects into one of the pre-defined object classes.

First a background model of the scene under consideration is constructed. Background subtraction is then carried out on *k* frames to generate *k* foreground masks which are intersected to produce static foreground objects which are likely to be either abandoned objects or static humans. The detected foreground regions are then classified into different categories of interest. Foreground regions identified as humans are discarded while for other static and potentially suspicious objects, the owner of the object is sought within a predefined neighborhood of the detected item. Presence of owner within the neighborhood leads to the assumption that the object is attended. In case the owner is not found, the system back tracks the video to the point while the object was still attended and tracks the owner from the point until the owner is no more in the camera view resulting in generation of an alarm.

The system is evaluated on a set of five video sequences from two standard data sets, PETS 2006 & 2007 and i-LIDS. The scenarios considered assume that each item of luggage has one owner and each person owns at most one item of luggage. The system

successfully identified all cases where the individuals leave the scene without their luggage. The tracking module however has limitations that it is not able to follow the owner in case of occlusion which may result in false alarms. The detection methodology being based on an unsupervised approach does not require prior training of objects to be detected, and hence successfully detects objects of all shapes, sizes and orientations. The proposed system can be extended to more complex scenarios which are true representatives of the real world situations. A multi camera network may also be considered in this regard providing information from different perspectives.

# DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

# DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,

a work of this magnitude would not have been possible

# ACKNOWLEDGEMENTS

There is no success without the will of ALLAH. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support.We would like to thank our supervisor Dr. Imran Siddiqi, for his help and motivation throughout the course of our project. Without his help we would have not been able to accomplish anything. He provided us with the opportunity to polish our technical skills and guided us into this area of learning.

# TABLE OF CONTENTS

# LIST OF TABLES

**Table** **Page No.**

## LIST OF FIGURES

**Figure** **Page No.**

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

In recent years, there has been a number of incidents where terror organizations have planted explosive devices in ordinary baggage to cause immense disruption in mass transportation networks and other areas of critical infrastructure. The threat of unattended baggage has led to increased vigilance amongst security personnel and the general public to ensure that unattended baggage is reported and investigated with utmost urgency. In conjunction with the introduction of enhanced CCTV, this has enabled an increase in the breadth and scope of data that can be collected at key locations. Unfortunately, this has not been matched by a corresponding improvement in the capabilities of systems to interpret and filter the data. This has remained the duty of trained human operators who often do not have the capacity to process the breadth of data that is received. Consequently, the increase in data availability has been met by an increase in the number of false alarms; situations where unattended baggage has been incorrectly considered a potential threat. Often, due to the pressure to act quickly, the situational data is only analyzed once a major event has occurred. This has resulted in unnecessary disruption to business operations, with associated cost implications and a lack of confidence regarding security procedures and equipment.

Building upon existing surveillance technology, the intelligent security camera (ISC) system will process goods that have been abandoned. At the same time, the system will identify the individual who left the goods and will utilize the surveillance network to determine the current location of that individual and track their followed path. ISC system will improve the efficiency of security personnel by automatically filtering out the major false alarms and therefore focusing their attention only on credible threats.

Existing systems are Nice Vision Video Analytics, Pets 2006 , AVSS2007 and SUBITO. The system is using dataset of Pets 2006, AVSS 2007, Pets 2007.

## 1.2 Objective

The developed system detects unattended luggage items from videos of public places, search for their owners and declares them abandoned in case the owner is not found. The system also classifies the detected objects into three main classes which are luggage, trolley and humans.

## 1.3 Scope

Abandoned luggage represents a potential threat to public safety. Identifying objects as luggage, identifying the owners of such objects, and identifying whether owners have left luggage behind are the three main problems requiring solution.

System will provide functionalities to detect unattended luggage item and search for its owner. After detection of left luggage it will be able to classify the detected object, based on defined classes.

System will work on off videos of Pets 2006, AVSS 2007 and Pets 2007 , with different events and scenarios. ISC System can be used at bus stations, airports, hospitals, and any crowded place where security is required. The ability to reliably detect suspicious items and identify their owners is urgently necessary in various venues such as airports and train stations.

## 1.4 System Overview

Detection of abandoned luggage and its classification task is divided into three stages. In the first stage static object is detected using background subtraction technique, which can either be luggage or human. If it is luggage then it is extracted. The second stage detects and tracks the owner. Human detection technique is used to detect human .If the owner is present near the detected luggage, the track is rejected and no further processing is performed. otherwise luggage is declared unattended and warning is generated. System waits for the owner for *t* seconds before declaring it abandoned and generate an alarm. In the final stage, abandoned luggage is classified into one of the defined classes, which are luggage , humans and trolley.

## 1.5 System Environment

ISC System is stand-alone automatic system. It will continuously capture the video and will generate alarm in case of unattended left luggage detection. Mainly system environment will include videos from public places.

Stakeholders of system are people at airport, bus-stations, hospitals or any other place where system is being installed. As customers different companies can install the system at their organizations for the sake of security. For the present ISC system videos of Pets 2006, Pets 2007 and AVSS 2007 have been used.

## 1.6 Software Tools And Resources

### 1.6.1 OpenCV

The OpenCV Toolkit, was originally developed by Intel, and later released under Open Source License for research purposes. It was developed for computer vision applications mainly focusing on real-time analysis and builds on Intel’s Integrated Performance Primitives (IPP). The IPP provides an extensive library of parallelizable optimized software functions specifically engineered for data processing applications. This toolkit was adopted as it provided an efficient basis for development, whilst still allowing low level control of the underlying data types and structures. For this implementation the most recent revision (1.0) is used.

The toolkit is cross platform (Windows, Linux and OS X), and exclusively coded in C. It has an active Developer Community, and user contributed User Manual (included on accompanying DVD). It comes accompanied with pre-built samples demonstrating aspects of computer vision such as Face Tracking, Hough Lines and Motion Tracking. Although the toolkit contains many low level functions, it also has a large collection of high order functions; some designed for the specific applications mentioned previously, others more abstract.

### 1.6.2 Visual Studio C++ 2008

Microsoft Visual C++ (often abbreviated as MSVC or VC++) is a commercial, integrated development environment (IDE) product from Microsoft for the C, C++, and C++/CLI programming languages. It has tools for developing and debugging C++ code, especially code written for the

Microsoft Windows API, the DirectX API, and the Microsoft .NET Framework.

### 1.6.3 PETS Datasets

The data-sets are multi-sensor sequences containing left-luggage scenarios with increasing scene complexity. The scenarios are filmed from multiple cameras and involve multiple actors.

We used our system on the 2 videos of PETS 2006 , 2 video of PETS 2007 datasets and 1 video of AVSS 2007.

### 1.6.4 I-LIDS Datasets

ILIDS is produced by the Home Office Scientific Development Branch (HOSDB) with Security Service funding and consists of real CCTV footage based initially on four different scenarios, Parked vehicles, abandoned baggage, Sterile Zone and doorway surveillance.

We used scenario of abandoned baggage and tested our system on 2 videos of I-LIDS dataset.

# CHAPTER 2

# SYSTEM REQUIREMENTS

# 2 Requirement Specification

## 2.1 User Interface Specification

System has an interface for end users, who will use the system in their organization. Interface is user friendly in all respects. Form has "start" and "stop" buttons for video capturing. Interface has windows to show output video having detected luggage. Figure 2.1 shows the interface diagram.


FIGURE 2.1: User interface Diagram

## 2.2 Limitations and Assumptions

We are using PETS[1] data set so according to the context the ISC system will have limited our scenario where each item of luggage has one owner and each person owns at most one item of luggage. We are assuming such a scenario having one luggage item and two persons entering the scene.

## 2.3 Functional Requirements

### 2.3.1 Definition Of Luggage

**Attended Or Unattended Luggage**

Luggage is own and to by a person who enters the scene with luggage until such point that the luggage is not in physical contact with the person (contextual rule). The luggage is attended by owner only when they are within distance x units of the luggage (spatial rule). A luggage item is unattended when the owner is farther than y units from luggage where(y>=x) (spatial-temporal rule)[1]

**Abandoned luggage**

Left-luggage in context of ISC system is defined as items of luggage that has been abandoned by their owner. [1]

### 2.3.2 Frames Extraction

Videos of Pets 2006 and 2007 are the first requirement of ISC system. Off video is given as input to the system. For further processing, frames are needed to be extracted from video. Since abandoned luggage is assumed to remain static for more than T consecutive seconds, a number of video frames are collected from the past T seconds.

### 2.3.3 Background Subtraction

Next step of system is background estimation. The background model is constructed using selected frames. The background model comprises the average of the selected frames, with a standard deviation calculated on each background pixel to consider the pixel variation. Background subtraction is then performed on *n* sample frames to produce *n* corresponding foreground images.[2]

### 2.3.4 Localization

The foreground-mask sampling attempts to localize static and possibly abandoned luggage items within the camera view.The intersection of *N* number of foreground masks is taken as the static foreground object mask *S*. All static foreground objects are assumed to be either humans or luggage items. Each foreground region in *S* is checked to determine whether it is a human via Haar classifiers.[2]

### 2.3.5 Selective Tracking

Selective tracking is performed for owner tracking. Temporal tracking and spatial tracking are two techniques to search for owner. Temporal tracking is based on time domain. If a luggage is unattended for 30 seconds, alarm will be generated.

Spatial tracking is based on space domain. If there is no person in surroundings of luggage up to specified time and no owner is found , it will be declared unattended.

### 2.3.6 Object Detection And Classification

After detection of static left object, next step is classification of object. Luggage is defined to include all types of baggage that can be carried by hand .Possible classes of luggage may be trunks, bags, rucksacks, backpacks, parcels, and suitcases. ISC system will work on three defined classes of objects, which are luggage, trolleys and humans. This classification is based on training data of the system. SIFT features are extracted for classification of objects. All these objects could potentially be threaten, for example a trolley may also contain a beg.

### 2.3.7 Alarm Generation

An alarm generation system is required for the system. It will generate alarm in case of left luggage.

## 2.4 Detailed Non Functional Requirements

### 2.4.1 Functionality

System must provide required functionality of luggage detection and classification properly and accurately.

### 2.4.2 Performance

System should perform operations within the defined time.

### 2.4.3 Availability

System will be available whenever required. Once system is installed, it must provide required functionalities all the time.

### 2.4.4 Reusability

The developed system could be used as a module and integrated into more comprehensive video surveillance sytems including facial recognition, tracking and recognition of objects etc.

If some other surveillance system is to be developed with needs of system operations, it is easy to reuse. Implementation of system is understandable. Functions and classes defined in system can be used in any other appropriate security software.

### 2.4.5 Security

Unauthorised access to the system and its data is not allowed . User need to be proper logged in to use the system. Backup copies of all Pets and AVSS dataset should kept by system to safe data from accidental or malicious damage.

### 2.4.6 Testability

System will be able to be tested in order to free it from faults. Different tests including beta testing is necessary in order to remove faults and make the software perform in accordance with the requirements specified.

### 2.4.7 Integratability

Separately developed components of the software will work correctly together in system. Modules of system must collaborate with each other in such a way to perform in way to be most useful.

# CHAPTER 3

# DESIGN AND ARCHITECTURE

# 3. System Architecture

## 3.1 Development Methods

ISC system is designed using object oriented approach. OO methods use an iterative and incremental layered approach to develop an object model of a system. The object model can be divided into 3 layers: logical, dynamic and implementation model. UML diagrams are created in Visual Paradigm to design the static and dynamic view of project. Logical model constitutes Use-Case and class diagrams. Dynamic model is demonstrated using Interaction Diagrams while the Implementation model contains packages, libraries, classes etc.

## 3.2 Architectural Strategies

The reason ISC employs the Object-oriented design is that this approach is more efficient and a faster way to develop. This technology cuts development time, overhead and provides reusability, reliability and maintainability for this project.ISC is important with respect to security purposes, so it should be highly reliable and maintainable to avoid generation of false alarms.

## 3.3 Component Diagram

ISC architecture shown below provides a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to components. The main purpose here is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together to provide the desired functionality. ISC system can be divided into five main modules

1. Interface
2. Object Detection
3. Owner Tracking
4. Classification
5. Alarm Generation

Figure 3-1 shows UML component diagram which is used to display the architecture.

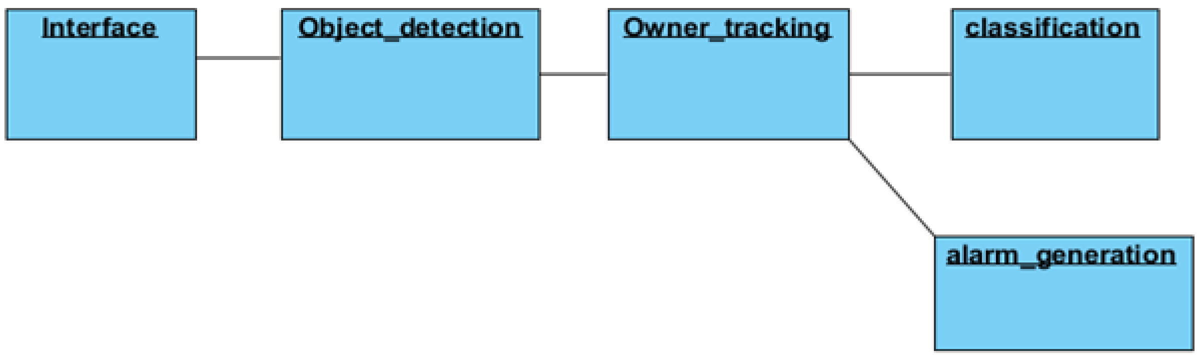



FIGURE 3-1: Component diagram

## 3.4 Detailed System Design

### 3.4.1 Logical View

Logical view contains class diagram and use case diagram. It describes the static behavior of system.

**1. Use-Case Diagram**

Figure 3-2 shows the interaction of user with the system.

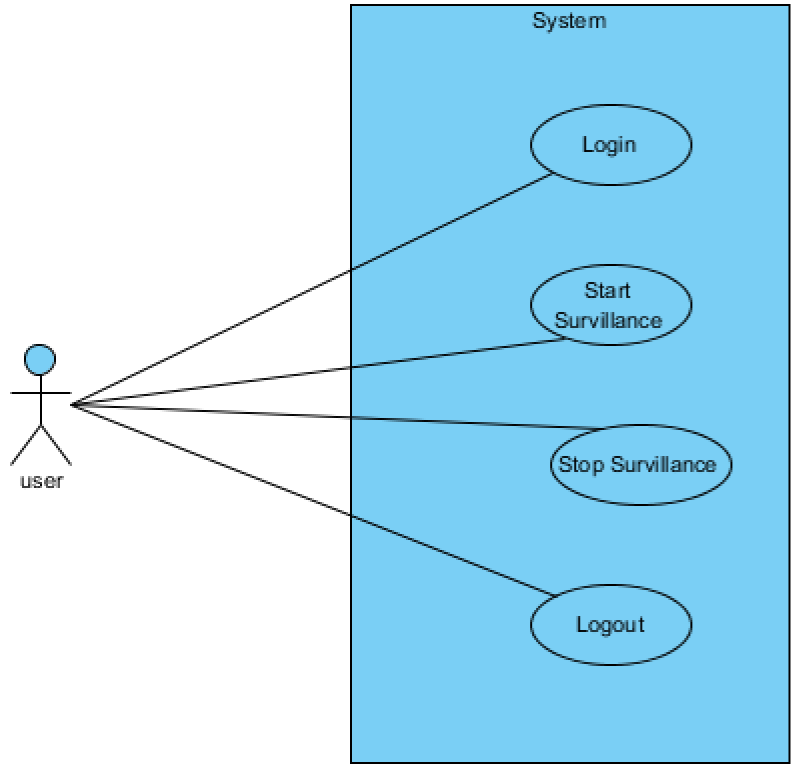



FIGURE 3-2: Use-Case Diagram

**2. Class Diagram**

Static behaviour of system in the form of classes is shown in Figure 3-3.It shows the classes and their relationship.

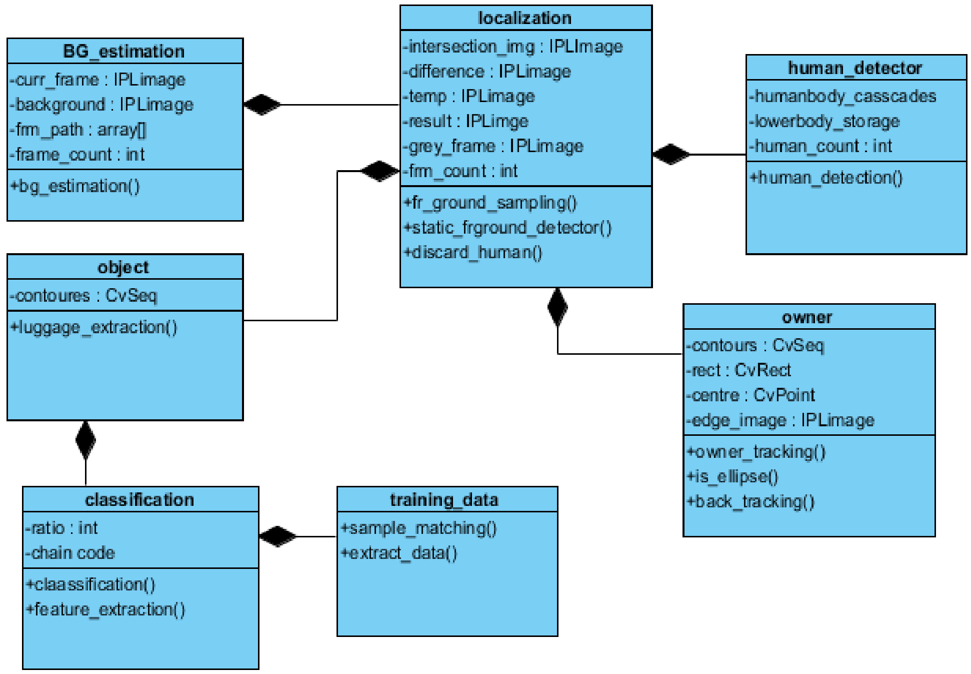



FIGURE 3-3: Class Diagram

### 3.4.2 Dynamic View

Dynamic view comprises of Interaction diagrams. Interaction diagrams are of two types:

**1. System Sequence Diagram**

System sequence diagram represent interaction between user and system. System is considered as a black box. Inner functionality of system is not shown. Figure 3-4 shows the System Sequence diagram. User log on the system, System verifies that user is authorized to access this system. Then user starts the surveillance and the automated system performs the desired functionality and generates alarm if there is abandoned luggage in the scene. Then user logs out of the system.

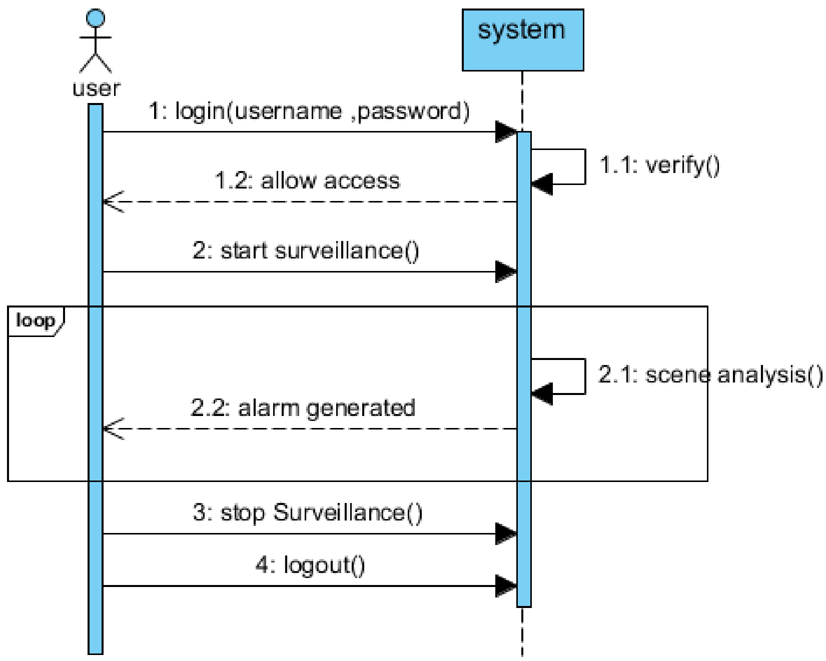



FIGURE 3-4: System Sequence Diagram

## 2. **Sequence Diagrams**

System is considered as a white box and the inner functionalities and how different calsses interact with each other using function calls or containership.Figure 3-5(a),(b) and (c) shows the interaction between various classes.

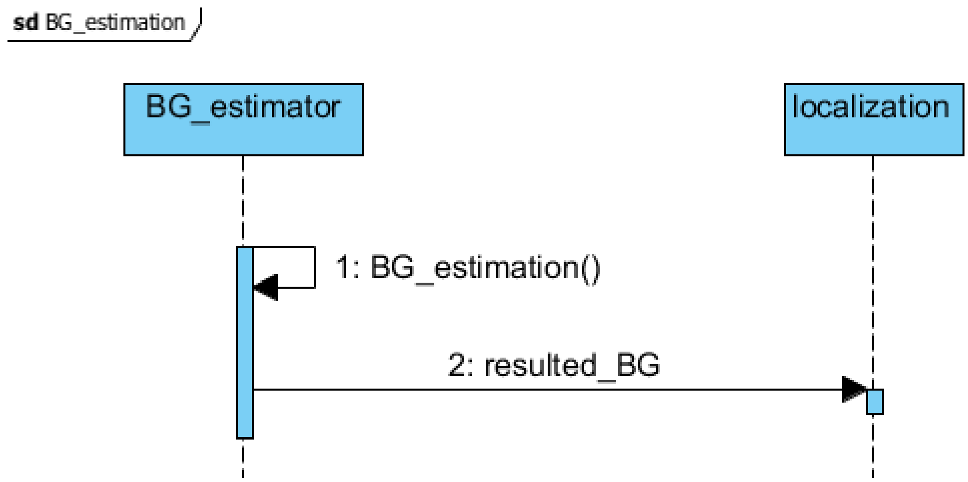



FIGURE 3-5(a): interaction between classes ‘BG_estimator’ and ‘Localization’.

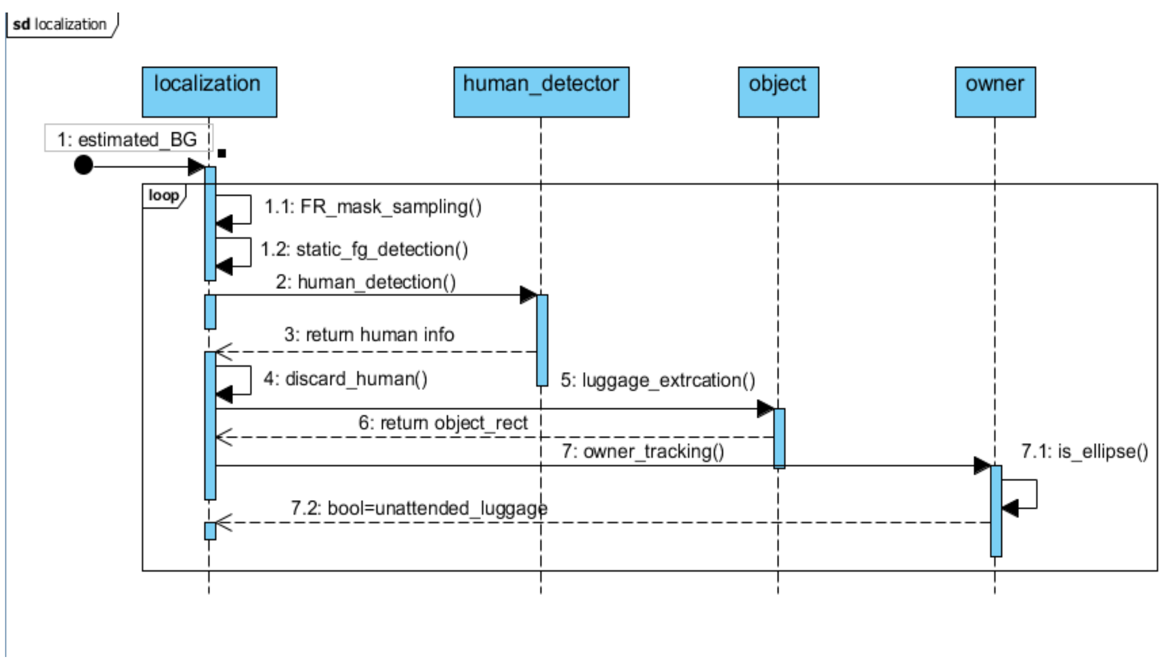



FIGURE 3-5(b): interaction between classes ‘Localization’, ‘human_detector’, ‘object’ and ‘owner’.

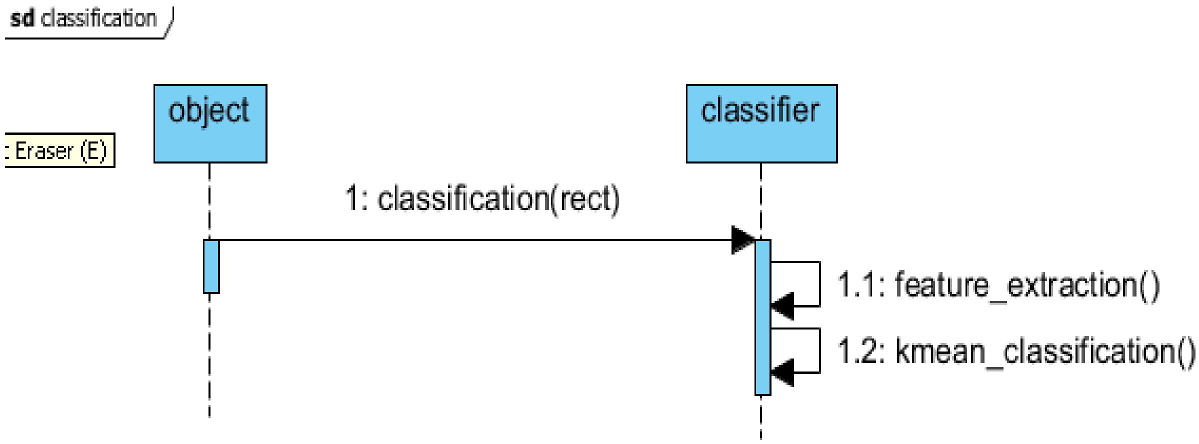



FIGURE 3-5(c): interaction between classes ‘object’ and ‘classifier’.

### 3. Communication Diagrams

They are similar to the sequence diagrams and describe the dynamic behaviour of system. Communictaion diagrams are shown in Figure 3.6(a)(b) and(c).

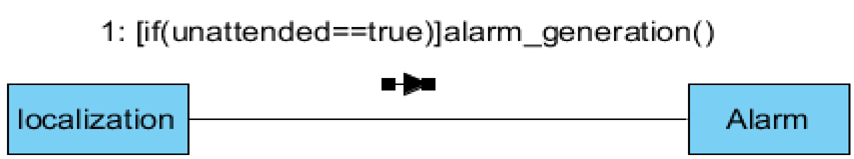



FIGURE 3-6(a): Communication between classes ‘Localization’ and ‘Alarm’.

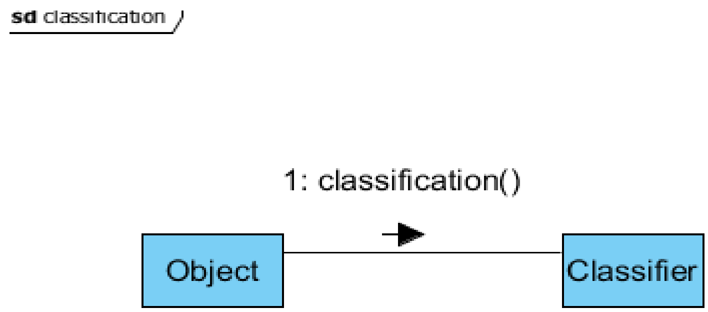



FIGURE 3-6(b): Communication between classes ‘object’ and ‘classifier’.

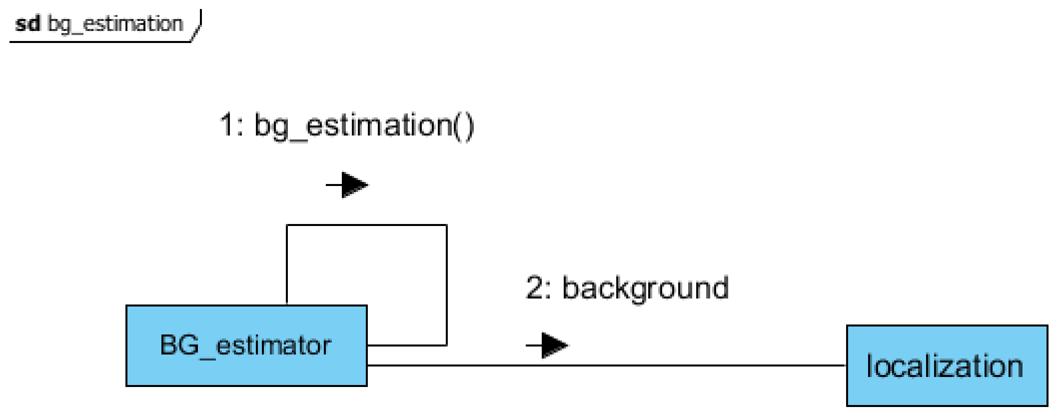



FIGURE 3-6(c): Communication between classes ‘BG_estimator’ and ‘localization’.

## 4. State Machine Diagram

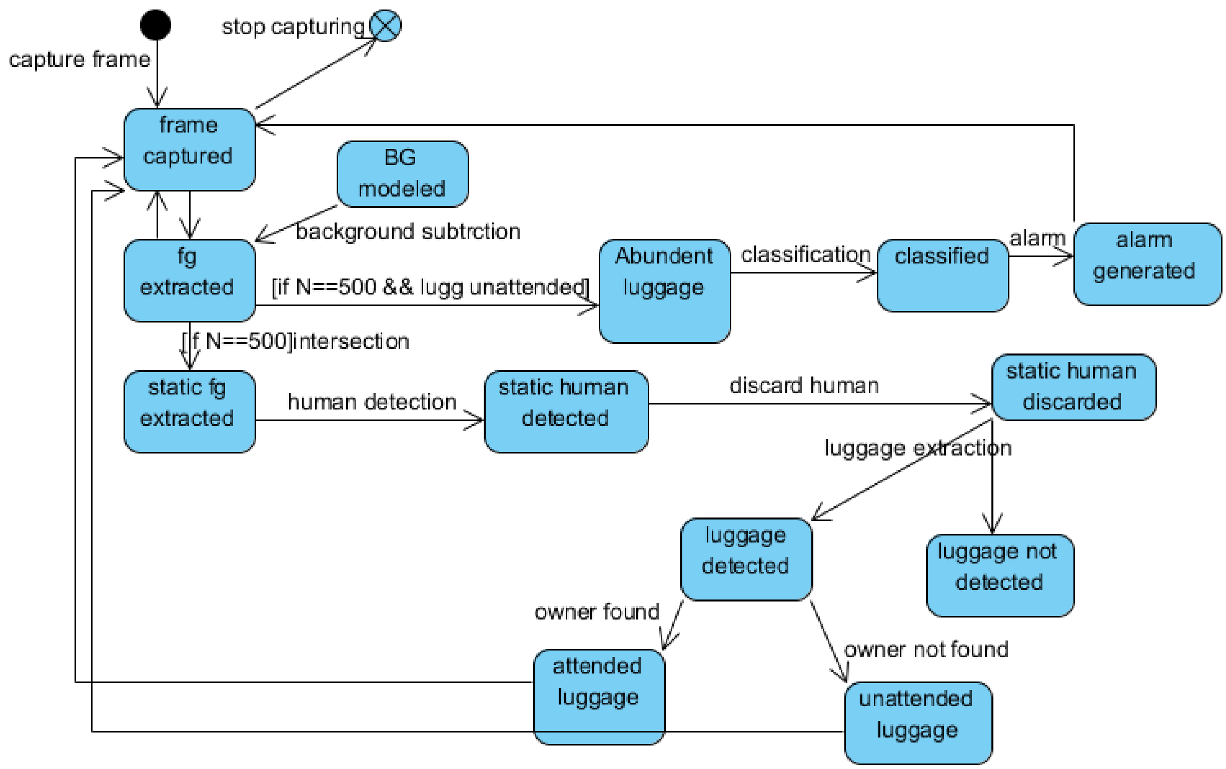



FIGURE 3-7: State Machine diagram.

### 3.4.3 Implementation View

Implementation view contains packages,classes,libraries etc. It represents the system in layered form. Package diagram is used to describe this view which is shown in Figure 3.7.ISC system is divided in three layers.

1. **Interface Layer**

   All the classes used for interfacing are placed in this layer. Classes describe the interaction of user with the system.

2. **Database Layer**

This layer contains the classes used for storing and maintaining of data used by the classes in application layer.

3. **Application Layer**

Also known as Business layer. Core functionality of the system is performed by the classes grouped in this layer.

Package diagram is used to describe this view which is shown in Figure 3-8.

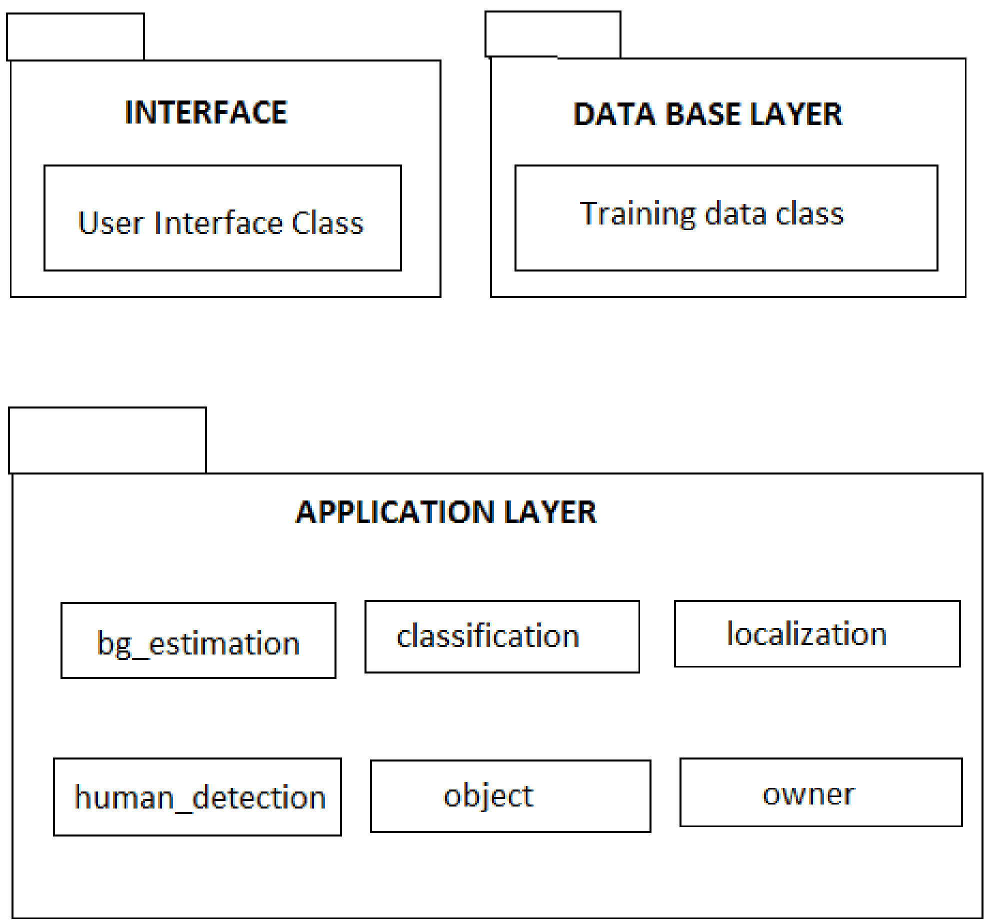



FIGURE 3-8: Package diagram

# CHAPTER 4

# IMPLEMENTATION

## 4.1 SYSTEM WORK FLOW

The detection of abandoned luggage and its classification is divided into three main stages. First stage localizes the luggage using background subtraction. Second stage detects object and owner using human detection technique and tracks the owner. And in the final stage classify the object into one of the predefined classes.

Video frames are processed using background subtraction which produces foreground mask samples. Foreground mask sampling localizes the static object. If the static object detected is human ,it is discarded else if it is luggage, owner is searched in its trajectory, if owner is not found then it is tracked and alarm is generated and luggage detected is classified using sift features. Classification is based on three main classes of trolley, beg and humans. Figure 4-1 shows the system work flow.

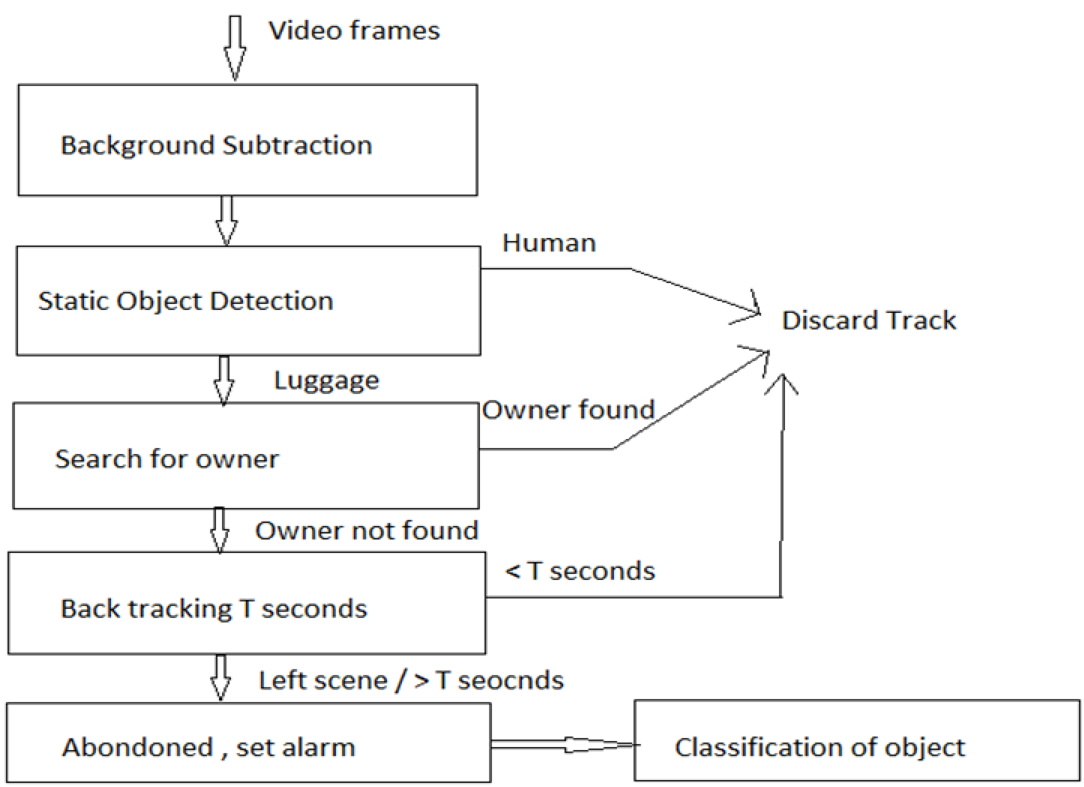



FIGURE 4-1: System work flow

## 4.2 Background Estimation

Background estimation is the process to detect a movement or significant differences inside the video frame, when compared to a reference, and to remove all the non-significant components (background)[24-27].

### 4.2.1 Introduction

Background Estimation is the first phase in detecting the abandoned luggage. Background subtraction is performed on *n* sample frames to produce *n* corresponding foreground images. Background subtraction works reasonably well when the camera is stationary and the change in lighting is gradual they also represent the most popular choice to separate foreground objects from the current frame[1,2].

### 4.2.2 Algorithms

For background estimation, we worked on three different methods which are described below:

1. **Frame Difference**

First is the Frame Difference algorithm. It compares the frame with the subsequent frame, therefore allowing the scene changes and updates[3].

$$\left| f_i - f_{(i-1)} \right| > Ts \tag{1}$$

F is the frame, $i$ is the frame number and T is the threshold and s is the threshold value.

**Results and analysis**

The results are not at all suitable for our project as it does not separate completely and accurately foreground from background. It generates a lot of noise. Figure 4-2 shows some results of Frame difference. Very sensitive to threshold T.

(a) (b)

FIGURE 4-2: The execution of algorithm.(a) and (b) shows the foreground pixels.

**2. Average Model**

The simplest background model assumes that every background pixel brightness varies independently, according to normal distribution.The background characteristics can be calculated by accumulating several dozens of frames, as well as their squares. That means finding a sum of pixel values in the location *S(x,y)* and a sum of squares of the values *Sq(x,y)* for every pixel location.

Then mean is calculated as

$$m(x, y)=S(x, y)/N \quad (2)$$

where $N$ is the number of the frames collected.

The above technique can be improved. First, it is reasonable to provide adaptation of background differencing model to changes of lighting conditions and background scenes, e.g., when the camera moves or some object is passing behind the front object. The simple accumulation in order to calculate mean brightness can be replaced with running average.

This method calculates weighted sum of two images. Once a statistical model is available, slow updating of the value is often required to account for slowly changing lighting, etc. This can be done by using a simple adaptive filter:

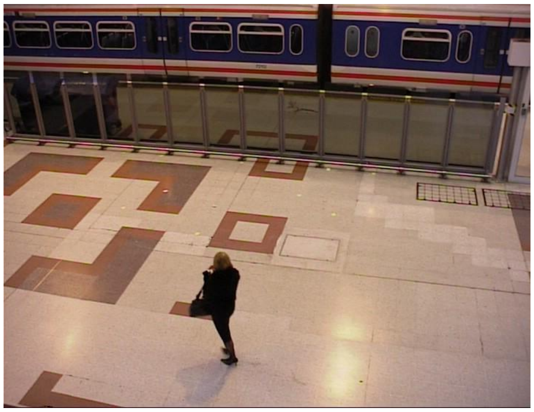
$$\mu t = \alpha y + (1-\alpha)\, \mu t\text{-}1 \quad (3)$$

where $\mu$ is the updated value,$0 \leq \alpha \leq 1$ is an averaging constant, typically set to a small value such as 0.05, and $y$ is a new observation at time $t$. When the function is applied to a frame sequence, the result is called the running average of the sequence.

**Results and analysis**

Background model is not constant,it changes with time.Requires high memory.

So taking into account the acceptable accuracy the Average model provides,we decided to use Average model in the project. Figure 4-3 shows the result of Average model.



(a) (b)

FIGURE 4-3: (a) Original image, (b) Foreground from Average model.

### 3. Mixture of Gaussian Model

The values of a particular pixel are modeled as a mixture of adaptive Gaussians. Multiple surfaces appear in a pixel. Lighting conditions change. At each iteration Gaussians are evaluated using a simple heuristic to determine which ones are mostly likely to correspond to the background. Pixels that do not match with the back-ground Gaussians are classified as foreground. Foreground pixels are grouped using 2D connected component analysis.

Using GMM first of all initial background is modeled over ***n*** number of frames. Over the time ***t*** different values appear for a single pixel in each frame. For example in an outdoor scenario , a pixel at location (***i* , *j***) may have different values from tree leaves, tree branches, and the building itself. Thus a multi-valued background is estimated using GMM. A histogram is constructed for each frame. Over the different frames due to varying background , each pixel has different values. Figure 4-4 represents three Gaussians for time $\boldsymbol{t_0}$, $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$.

Due to day and night changes, pixel value varies in different frames. Thus three backgrounds are modeled using three different Gaussians. Each Gaussian constructed, has mean value $\boldsymbol{\mu}$ and variance $\boldsymbol{\alpha}$ . If a new pixel value lies within the variance range upto $\pm 3\alpha$ , it is considered to be the part of that Gaussian distribution. And if the value of new pixel is far away from all Gaussians , it is considered to be the part of foreground. If the value of the pixel doesn't change for certain time , a new Gaussian is drawn for this pixel, and it become the part of background. Thus GMM provides constant updating of background model.

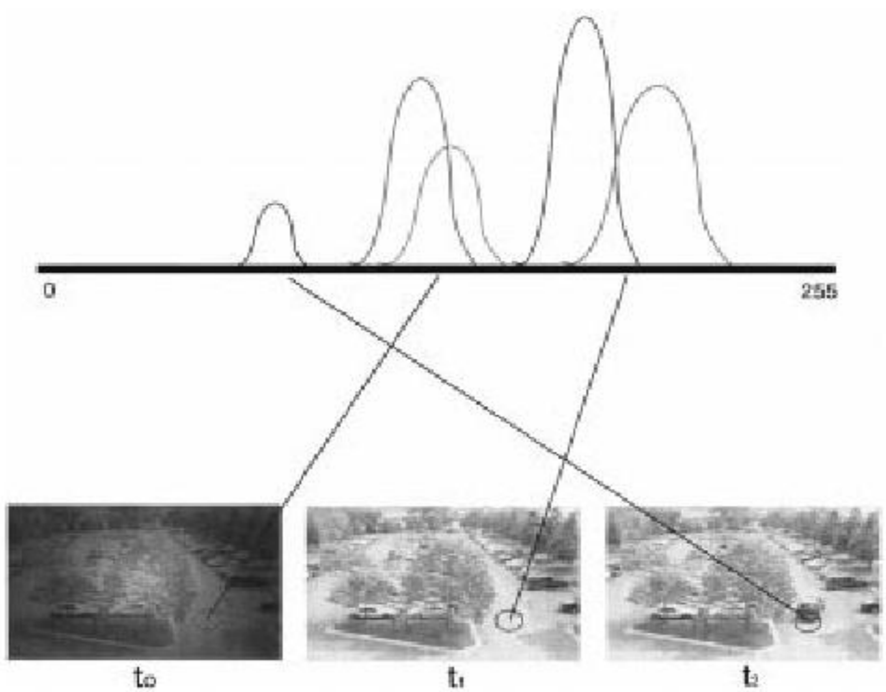



FIGURE 4-4: (Gaussian) Distributions

### Background Model Estimation

The Gaussians with the most supporting evidence and least variance correspond to the background.The Gaussians are ordered by the value of ω/ơ (highsupport and less variance will give a high value).Then simply the first B distributions are considered as the background model:

$$B = \operatorname{argmin}_b\left(\sum_{i=1}^{b} \omega_i > T\right) \tag{9}$$

where T is minimum portion of the image which is expected to be background.

After background model estimation red distributions become the background model and black distributions are considered to be foreground.

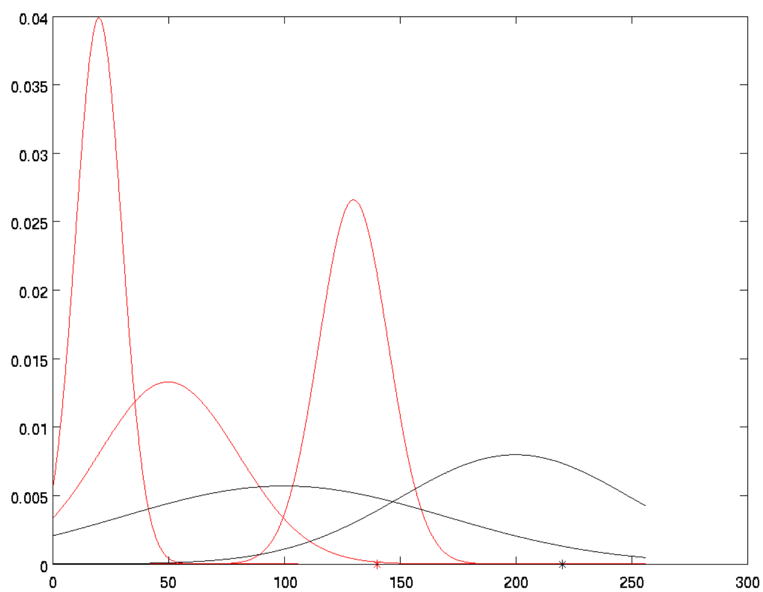



FIGURE 4-5 : Background and foreground distributions.

**Results and analysis**

Gaussian model is preferable in case of outdoor scenarios where there is high light intensity fluctuations i.e. day and night time. Gaussian model continuously updates the background and so the light fluctuations become the part of the background instead of considering these changes as false foreground. But in our system indoor scenario is used, where light intensity is almost constant. And also system doesn't need to update the background, because we are dealing with static objects. [4]. Fig4-6 shows the result of Gaussian model.

(a) (b)

FIGURE 4.6: (a) original image, (b) Gaussian model.

## 4.3 Foreground Mask Sampling

After background estimation, next step is to calculate foreground. Foreground mask sampling localizes and separates the object of interest from the complex environment.

### 4.3.1 Background Subtraction

After background estimation, Background subtraction is then performed on *N* frames to produce *N* corresponding foreground images. Pixels of static region appear as foreground for *k* seconds, foreground mask samples $M_{k(i,j)}$ are collected in $F_k$ no of frames[2].

$$M_{k(i,j)} = \begin{cases} 1, & |F_{k(i,j)} - B_{(i,j)}| \geq w_{(i,j)} .std_{(i,j)} \\ 0, & |F_{k(i,j)} - B_{(i,j)}| < w_{(i,j)} .std_{(i,j)} \end{cases} \tag{10}$$

Where $k = 1$ to $N$, and $w(i,j)$ is the weight of the pixel at location $(i,j)$.More weight is given to the lower pixels as compared to upper pixels. Most security cameras are deployed in a look down position, due to which lower part of image appears larger and closer, so they have a better resolution as compared to objects in upper part where both the image quality and resolution is not as good as shown in Figure 4-7.

FIGURE 4-7: PETS 2006 video data set. The surveillance camera is looking down, causing the lower body to appear larger than upper body.

The weight is calculated as a function of row to consider the gradual change in image in vertical direction.

$$w(i,j) = [c/h \cdot i \cdot w] \tag{11}$$

Where *h* is the height of image and *W* is the weight of top most pixels. *w* (*i*,*j*) increases the threshold level of lower part of images where the screen resolution is high and decreases the threshold value where image quality and resolution is low. Thus balancing the resolution of images.

Stationary foreground mask *S* is then obtained by taking the intersection of foreground mask samples. Each pixel is determined to be a part of foreground.[2]

$$S = M1 * M2 * M3 * M4 * \ldots\ldots * Mk \qquad (12)$$

Where M1….M*k* are the foreground mask samples.

We then apply smoothing filter to remove noise. The portion of image which is selected as foreground is represented by white pixels. Figure 4-8 shows some foreground mask samples.

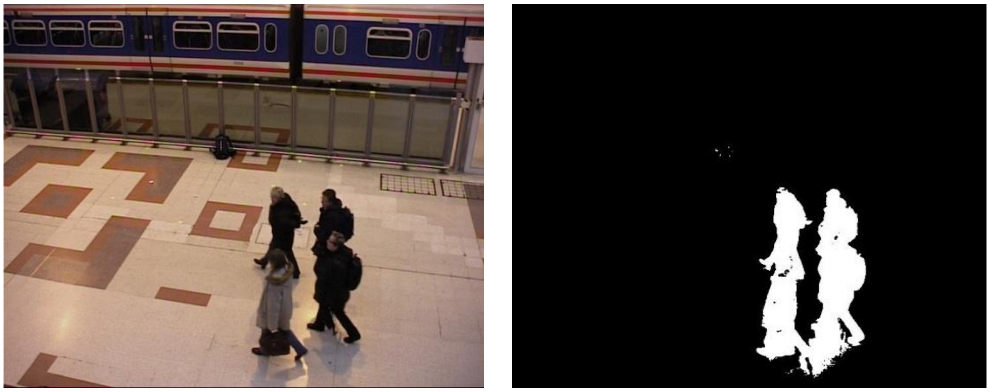


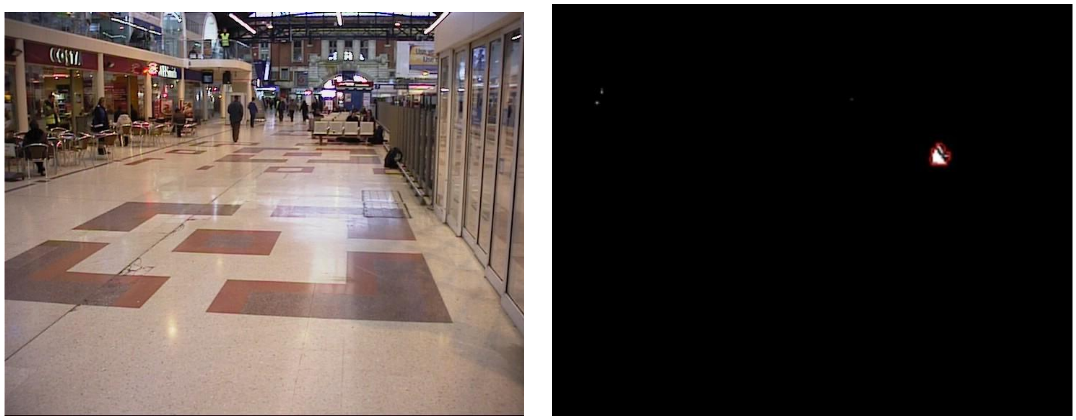
FIGURE 4-8: first column shows original image, and second column shows *n* foreground mask samples when applied to *n* frames.

### 4.3.2 Static Object Detection

This foreground static region either contains static human being or luggage. The region is then analyzed to differentiate between humans and luggage. Human detection algorithm is used to detect a human which is explained in Section 4.4.

If the region is identified as human, we will discard it.

In the case of unattended luggage, we'll search for owner. The appearance-based model is not used in locating suspicious luggage items, and thus can deal with luggage of any color and shape and is not affected by different viewing angles[5].using haar classifiers humans are detected and discarded. Thus luggage is extracted. Luggage extracted is shown in Figure 4-9.



(a) (b)

FIGURE 4-9: (a) Original image (b) Luggage extracted from the scene.

## 4.4 Human Detection

All foreground static objects are assumed to be either humans or luggage items. To distinguish humans from arbitrary objects, several well known algorithms are used. Human detection is used to search for owner[6-14,17-22].

### 4.4.1 Algorithms

1. **Face Detection Using Skin Colour**

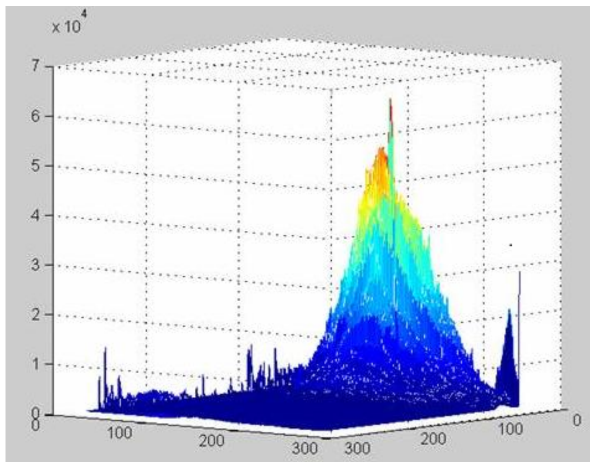
Face detection consists of three steps. In the first step, pixel in given image is classified as skin pixel or non-skin pixel. In the second step, different skin regions in skin detected image are identified using connectivity analysis. In the final step, each detected skin region is examined whether it is a face or not using two parameters, height to width ratio of the skin region and the percentage of skin in the rectangle defined by the height and width.[6]

**Skin pixel Classification**

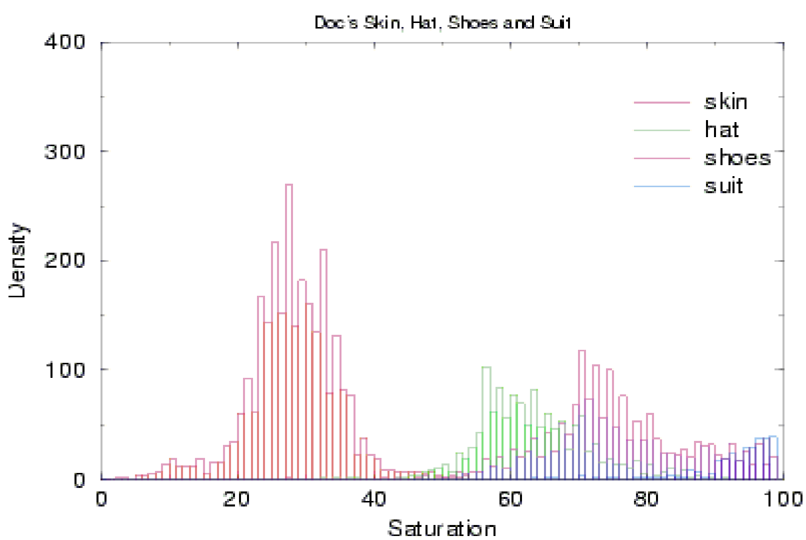
HSV normalized RGB and YCrCb colour spaces are used to detect skin.HSV gives best results in detecting the skin pixel than RGB and YCrCb. Skin colour can be represented by histograms or Gaussians.

In HSV, H stands for hue, which describes the shade of the color, S stands for saturation, which describes the purity of color and V stands for value component, which describes the brightness varies from 0 to 1 on a circular scale which means colours represented by H=0 and H=1 are the same. S varies from 0 to 1, 1 representing 100 percent purity of the colour. H and S scales are partitioned into 100 levels and the colour histogram is formed using H and S.

For each pixel in training data, normalized histogram of H and S is plotted. Figure 4-10 (a) and (b) shows the 3 and 2 dimensional obtained histogram.




(a)




(b)

FIGURE 4-10: HS histograms of skin pixels (a) 3d view , (b) 2d view[6]

The height of histogram is proportional to the probability of skin pixel. Threshold between 0 and 1 is used to classify any pixel as skin or non-skin pixel. This threshold known as a skinthreshold classifies the pixel as skin if the height of H and S values exceeds the threshold and vice versa. Only skin pixels are shown in skin detected image.

**Connectivity Analysis**

Skin detected image shows the skin pixel. But still it cannot be declared that whether it belongs to face or not. For this purpose we group the pixels using 8-connectivity neighborhood-e if the given skin pixel has another skin pixel in its 8 neighbors, then both pixels belongs to the same region. This is done by calculating the centroid, height and width of the region and the percentage of skin in the rectangular area defined by the above parameters. The centroid is calculated by the average of the coordinates of all the pixels in that region.

If the height to width ratio falls within the range of golden ratio which is

$$( 1+\sqrt{5}) / 2 \pm \text{tolerance} \qquad (14)$$

If the percentage of skin is higher than percentagethreshold,then that region is considered a face region.This algorithm may detect false pixel if there are colors in the image which resemble skin but are not skin pixels.

Various thresholds used in algorithm are shown in table 4-1

| THRESHOLD | VALUE |
| --- | --- |
| SKINTHRESHOLD | 0.1 |
| PERCENTAGETHRESHOLD | 55 |
| TOLERANCE | 0.65 |

Table 4-1: Thresholds and their values[6]

**Analysis**

Skin detection algorithm did not give good results as the human bodies in the scene are small enough to be detected by face and hand skin colors and the images can have other non-human objects with skin like color.

Generic color models are usually sensitive and environment-specific. The color model needs to be adapted regularly as the human's color varies with environmental conditions [7].

**2. Motion Detection**

Each frames contains different colour pixels in every row and column. Subtracting two consecutive frames gives the difference between the two images. The differences of pixel value are considered as motion region and a threshold value is used to decide the output image. If the pixel value is greater than the threshold value, threshold is set to 255 and set to 0 if pixel value is less than threshold value. After thresholding, white pixel represent the moving areas.

$$\Delta = abs\ (\text{ frame}N - \text{frame}N+1) \quad (15)$$

Segmentation is used to divide the image into meaningful regions to represent specific area of an image. It eliminates non-motion region areas. Segmentation not only fastens the process by reducing the scanning window in such region but also reduces the false alarm percentage when the segmentation is done. The edge of white pixel are then combined finding the connected component. Thus segmenting out the region in rectangular form . It segments out two or more rectangles in an image to show that there are some objects moving in that particular image. Haar cascades are then used to detect the humans in the segmented region .

### 3. Haar Cascades

Haar-like features are the digital image features used in object detection. Viola and Jones used idea of Haar wavelets and developed the Haar-like features.[8]

Human detection is much more efficient when it is based on detection of features that gives programs the information about humans. Haar like features encodes the contrasts between the regions in the human face. It consists of adjacent rectangular regions at a specific location in a detection window, adds the pixel intensities and then calculates the difference between them. For example, to detect humans, image database with human beings are used.As in all faces the region of the eyes is darker than the region of the cheeks. Therefore a common Haar feature for face detection consists of set of two adjacent rectangles that lie above the eye and the cheek region.

The position of these rectangles is defined relative to a detection window that acts like a bounding box to the human beings.

To search the object in the whole image search window is moved across the image and every location is checked using the classifier. If object is detected in the region, classifier outputs 1 and 0 otherwise. Classifier can be easily resized and can be used to detect object of different sizes.

The word “cascade” in the classifier means that classifier consists of many simpler stages that are applied to the region of interest until all the stages are passed.

Upper-body , lower-body and head-shoulder cascades are used for detection.full-body detector is also used.Figure 4-11 shows the body parts definition.

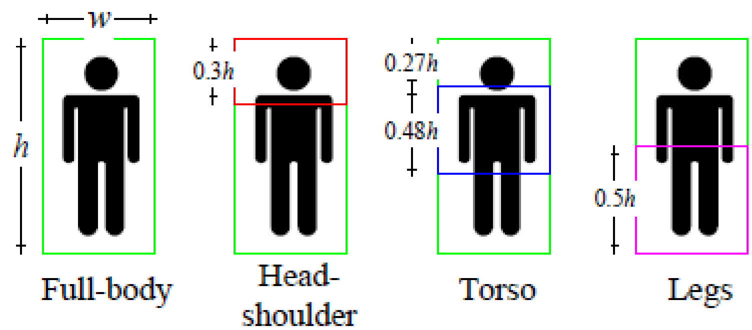



FIGURE 4-11: The definition of body parts[8]

**Inter-object Occlusion**

We have assumed that humans walk on the ground and camera captures the image by looking down which is true for most surveillance systems[8], as shown in figure 4.12.

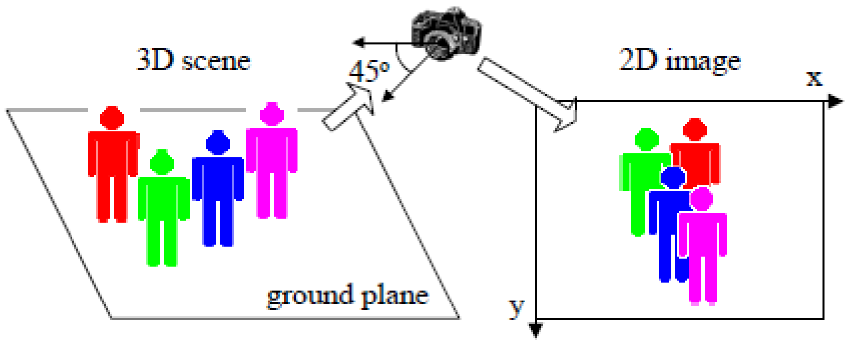



FIGURE 4-12: Assumption

If a human in the image is visible then at least his/her head is visible, in this case head-shoulder cascades are usefull.

In the case of ooclussion, results of upper-body and lower-body cascades are combined for detection and to avoid false alarms generation.

**Results and Analysis**

For the crowded scenes the performance of full-body and lower-body detectors decreases, as lower-body is more likely to be occluded. The combined detector gives the best results.

Based on analysis and results human is detected at least once in every fourth frame. if some features are not detected in one classifier, other is going to detect these features. In our experiments the accuracy with which the human body is detected is 75%, which means out of every four humans three are detected.Figure 4-13 shows the results of human detection using Haar cascades.

(a) (c)

(b) (d)

FIGURE 4-13: (a),(b),(c) and (d) shows the results of Human detection using Haar cascades. Full-body and lower body.

## 4.5 Owner Tracking

All foreground regions contain either humans or static object. Each foreground regions are identified as humans using different human detection algorithms as explained in Section 4.4. A local search region is constructed around the detected luggage to see whether its owner is in close proximity in the present frame at time $t$. If the owner is found, the region is again discarded because the owner exhibits no intention of abandoning the luggage. However, if the owner is not located near the luggage, algorithm waits for $t$ seconds and after that it gives a warning. It will wait for ($t$+ $\Delta t$) seconds if the owner is still not in the region, it will generate an alarm [1].

Owner is searched using human detection algorithm.

$\Delta t$ is set to 30 seconds in this case, assuming that when luggage was first detected, owner must have been in the near proximity of the luggage. This assumption is valid because if the owner has been absent for some time, foreground-mask sampling technique will detect the isolated luggage item faster. Owners who abandon their luggage with criminal intention would generally want to avoid attention and are unlikely to loiter. Instead they will remain constantly with their luggage prior to abandonment .If multiple people surround the abandoned luggage, person closest to the luggage is assumed to be the owner. Following rules are defined for declaring luggage unattended and then abandoned [1].

**Spatial rule**

In spatial rule a two circles are drawn around luggage in some region. Owner is tracked in this region. The Figure 4-14 shows a person within the specific region standing close to his  luggage. In this situation no alarm should be raised by the system.

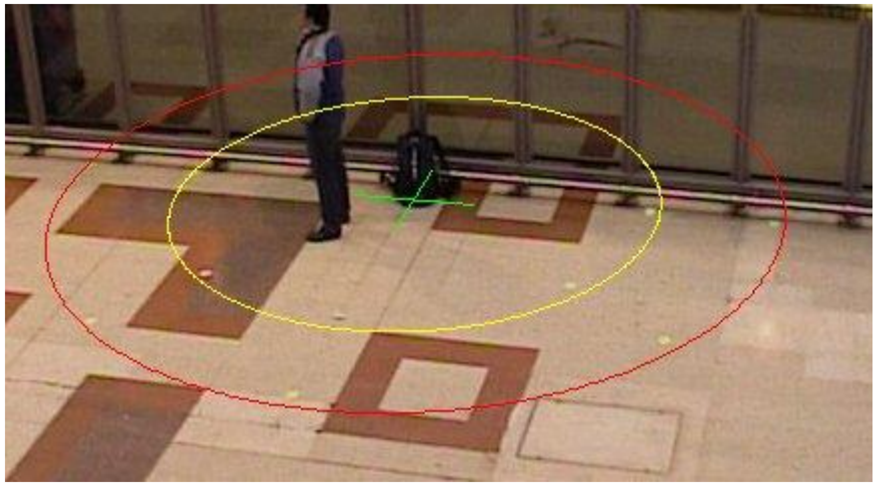


FIGURE 4-14: Spatial rule

**Temporal rule**

An item of luggage that has been left unattended by the owner for a period of *t* consecutive seconds in which time the owner has not re-attended to the luggage, nor has the luggage been attended to by a second party (instigated by physical contact, in which case a theft / tampering event may be raised). The Figure 4-15 shows an item of luggage left unattended for *t* (=30) seconds, at which point the alarm event is triggered.

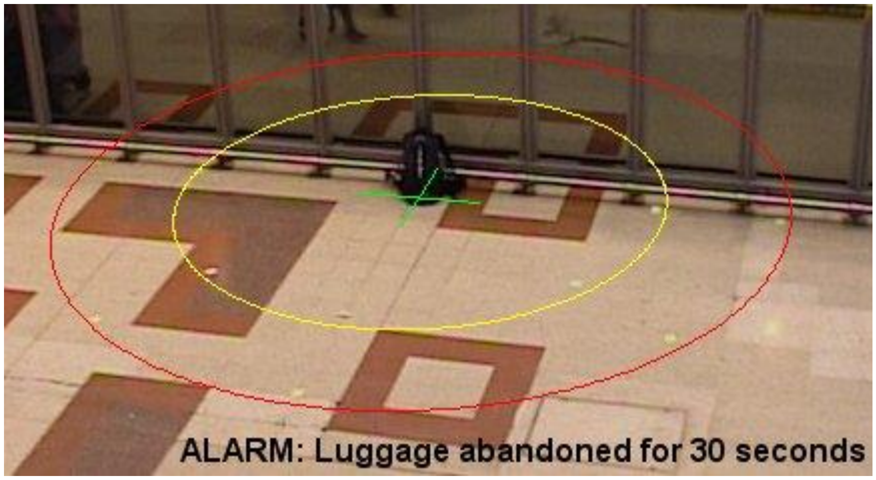



FIGURE 4.15: Temporal rule

**Warning event**

We defined two circles around luggage. If the owner is out of the circumference of first circle only warning is generated. This zone is defined to separate the detection points of the two states, reducing uncertainties introduced due to calibration / detection errors in the sensor system etc. The Figure 4-16 shows a person crossing the line of first circle, but within the second circle. In this scenario the system can be set up to trigger a warning event, using a rule similar to the spatio-temporal rule.

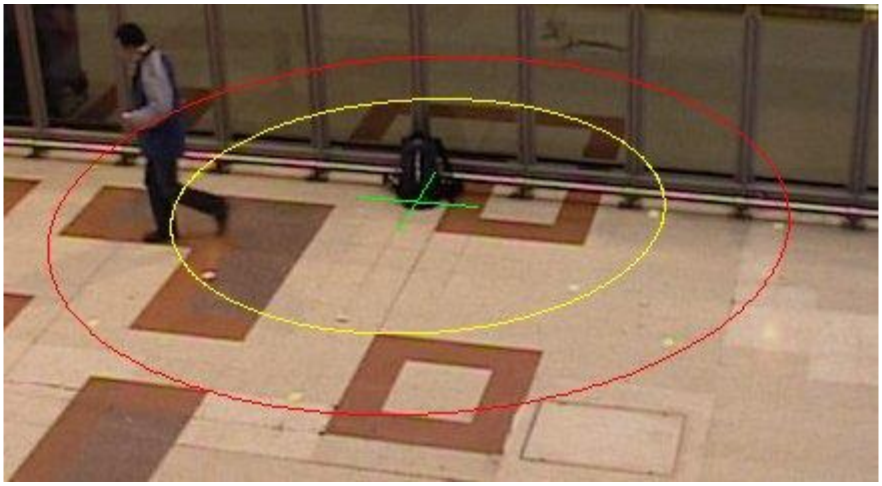


FIGURE 4-16: Warning event

**Alarm event**

A luggage item is unattended when the owner is out of both circles around the luggage. The Figure 4-17 shows a person crossing the circumference of second circle. In this situation the system should use the spatio-temporal rule to detect whether this item of luggage has been abandoned.

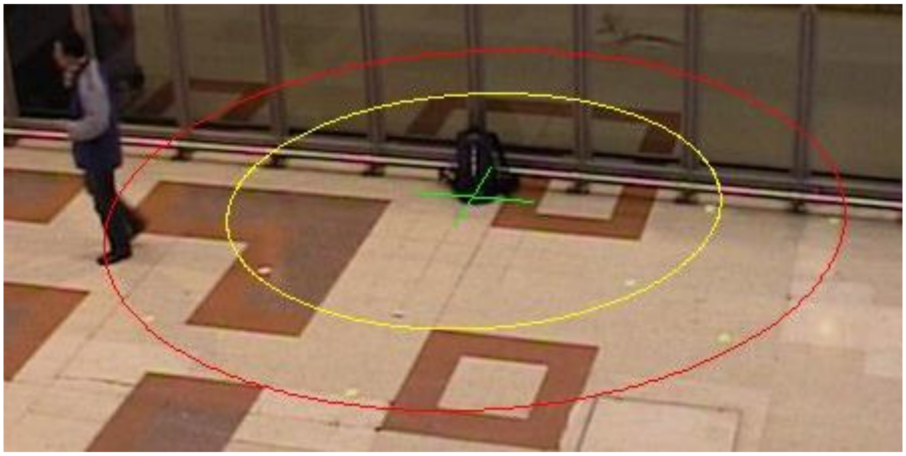


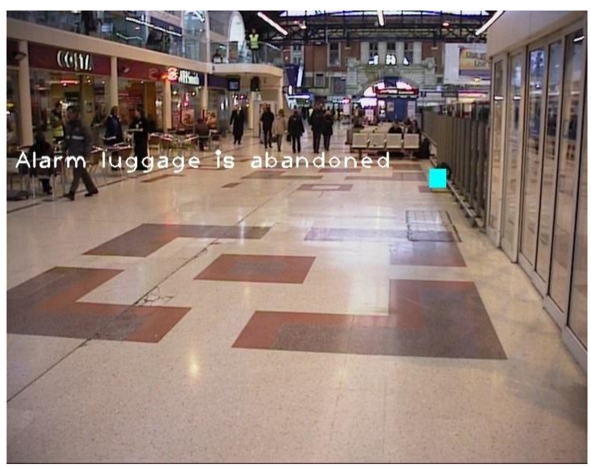
FIGURE 4-17: Alarm event

### 4.5.1 Results

Figure 4-18 shows the results of owner tracking obtained using Spatial and Temporal rules defined above.

(a)

(b)




(c)

FIGURE 4-18: (a) Owner found, (b) Owner not found (c) Luggage Declared Abandoned

## 4.6 Abandoned Object Classification

In this phase, abandoned object is classified into one of the predefined classes. Various features are used to discriminate different objects, and then classifier is built on these features.

Classification is based on three main approaches [31].

1. Shape-based
2. Motion-based
3. Combined shape-motion

We have trained our system with 3 classes: trolley, bag and person.

### 4.6.1 Feature Extraction

Since abandoned object is static, we used only shape-based features. We have used features which are more reliable and can easily distinguish objects known as good features. SIFT features are used for extracting features.

**Sift Features**

SIFT transforms image data into scale-invariant coordinates relative to local features. Each of these feature vectors is invariant to any scaling, rotation or translations of the image.Set of features are generated by four step algorithm [28].

1. **Scale-Space Extrema Detection**

   It used to detect stable key point locations in the scale-space. Scale-space is defined by the function.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad [16]$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image.

**2. Key point Localization**

Stable key points are selected. This is done by calculating the Laplace and key points with value less then threshold are eliminated from the list.

**3. Orientation Assignment**

Each key point is assigned an orientation based on its gradient magnitude and by forming a histogram of the sample points.

**4. Key point Descriptor**

The local gradient data is used to create key point descriptors. Processing key points result in a feature vector containing 128 elements. These resulting vectors are known as SIFT keys.

SIFT keys are then applied on the images. Figure 4.19 shows examples of SIFT keys detected in a number of images.

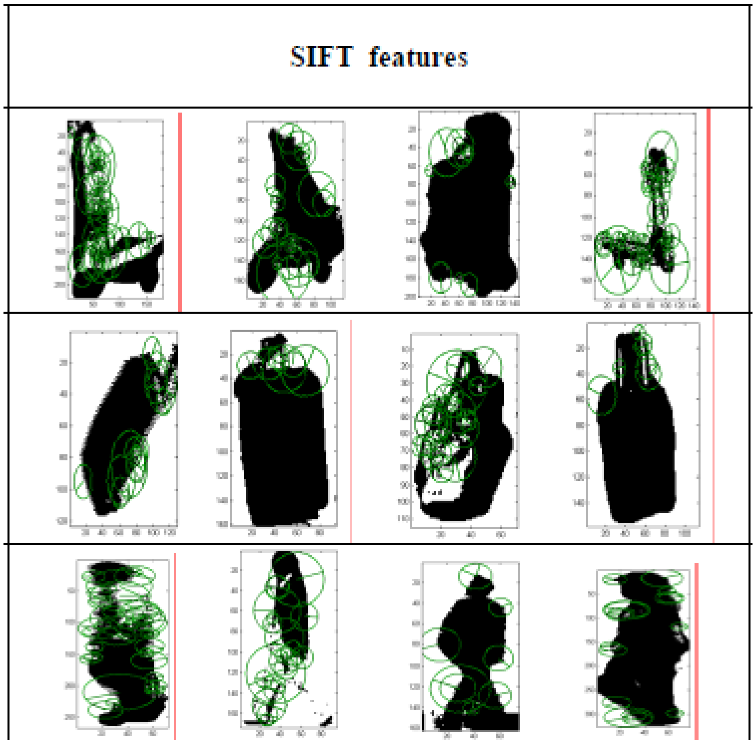



FIGURE 4-19: Examples of features detected in a number of images:

Trolley (1st row), bag (2nd) and person (3rd) [31]

### 4.6.2 Results


FIGURE 4-20: Images for beg and trolley classification

# CHAPTER 5

# RESULTS AND EVALUATION

## 5.1 Background estimation

Evaluation results shows that processing time for frame subtraction is 17.2 ms , for Average model it is 40.7 ms . Gaussian model has processing time 98.6 ms, which is far off greater than average model. Each method was tested and the execution time was recorded in given table below.

| Serial no | BG Estimation Model | Average time (milisecond ) | Relative results |
|---|---|---|---|
| 1 | Frame Subtraction | 17.2 | T |
| 2 | Average Model | 40.7 | 2.4t |
| 3 | Gaussian Model | 98.6 | 5.7t |

Table 5-1: Results for background estimation

## 5.2 Haar Classifiers

<table>
<tr><th>Serial no</th><th colspan="2">Dataset</th><th>Test images</th><th>Humans actually present(x)</th><th>True positive (y)</th><th>False positive (z)</th><th>Recall (y/x)</th><th>Precision y/(y+z)</th></tr>
<tr><td rowspan="5">1</td><td rowspan="10">Pets2006</td><td rowspan="5">Video1</td><td>1</td><td>6</td><td>4</td><td>0</td><td rowspan="5">77%</td><td rowspan="5">87%</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>3</td><td>4</td><td>3</td><td>0</td></tr>
<tr><td>4</td><td>6</td><td>5</td><td>0</td></tr>
<tr><td>5</td><td>7</td><td>6</td><td>2</td></tr>
<tr><td rowspan="5">2</td><td rowspan="5">Video2</td><td>1</td><td>4</td><td>4</td><td>0</td><td rowspan="5">77%</td><td rowspan="5">77%</td></tr>
<tr><td>2</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>3</td><td>2</td><td>1</td><td>1</td></tr>
<tr><td>4</td><td>3</td><td>2</td><td>0</td></tr>
<tr><td>5</td><td>1</td><td>1</td><td>1</td></tr>
</table>

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3 | iLids_AVSS | 1 | 2 | 1 | 1 | 73% | 67% |
| | | 2 | 3 | 2 | 1 | | |
| | | 3 | 3 | 2 | 1 | | |
| | | 4 | 1 | 1 | 1 | | |
| | | 5 | 2 | 2 | 0 | | |
| 4 | Pets2007 | 1 | 2 | 1 | 0 | 57% | 69% |
| | | 2 | 2 | 2 | 0 | | |
| | | 3 | 5 | 2 | 2 | | |
| | | 4 | 1 | 1 | 2 | | |
| | | 5 | 6 | 3 | 0 | | |
| OVERALL RESULTS | | | | | | 71.2% | 77% |

Table 5-2: Results for Haar classifiers

Results shows that haar classifiers will detect human beings with overall precision of 77%. True positive rate or recall of the system is 71.2%.

## 5.3 Classification

| Sample no | Query Instance | Classified as | Binary Result | 10 Nearest neighbors (Lug,Trolley,Human) |
|---|---|---|---|---|
| 1 | Lug | Lug | True | (6,1,3) |
| 2 | Lug | Human | False | (3,2,5) |
| 3 | Human | Lug | False | (6,0,4) |
| 4 | Human | Human | True | (2,3,5) |
| 5 | Human | Human /lug | Ambiguous | (5,0,5) |
| 6 | Lug | Lug | True | (7,2,1) |
| 7 | Human | Human | True | (4,0,6) |
| 8 | Human | Human | True | (4,1,5) |
| 9 | Human | Human | True | (0,1,9) |
| 10 | Human | Human | True | (3,3,4) |
| 11 | Lug | Lug | True | (7,2,1) |
| 12 | Human | Lug | False | (7,1,2) |
| 13 | Human | Human/lug | ambiguous | (5,0,5) |

| | | | | |
|---|---|---|---|---|
| 14 | Lug | Lug | True | (9,0,1) |
| 15 | Lug | Lug | True | (8,1,1) |
| 16 | Human | Lug/human | ambiguous | (4,2,4) |
| 17 | Lug | Human | False | (4,1,5) |
| 18 | Human | Lug/human | ambiguous | (4,2,4) |
| 19 | Trolley | Trolley | True | (1,9,0) |
| 20 | Trolley | Trolley | True | (0,10,0) |
| 21 | Human | Human | True | (4,0,6) |
| 22 | Human | Lug | False | (5,1,4) |
| 23 | Trolley | Trolley | True | (4,6,0) |
| 24 | Trolley | Trolley | True | (2,8,0) |
| 25 | Trolley | Trolley | True | (0,10,0) |
| 26 | Trolley | Lug/trolley | ambiguous | (4,4,2) |
| 27 | Trolley | Lug/human | False | (4,2,4) |
| 28 | Trolley | Trolley | True | (4,5,1) |
| 29 | Trolley | Trolley | True | (2,6,2) |
| 30 | Trolley | Trolley | True | (3,4,3) |
| 31 | Trolley | Lug | False | (5,3,2) |
| 32 | Trolley | Trolley/lug | Ambiguous | (4,4,2) |
| 33 | Trolley | Lug | False | (5,2,3) |
| 34 | Trolley | Lug | False | (6,3,1) |
| 35 | Trolley | Lug | False | (7,2,1) |
| 36 | Lug | Lug | True | (6,0,4) |
| 37 | Trolley | Trolley/human | Ambiguous | (2,4,4) |
| 38 | Lug | Lug | True | (7,1,2) |
| 39 | Lug | Lug | True | (6,2,2) |
| 40 | Lug | Lug | True | (6,2,2) |
| 41 | Lug | Lug | True | (7,2,1) |
| 42 | Lug | Lug | True | (7,3,0) |
| 43 | Lug | Human | False | (3,2,5) |
| 44 | Lug | Lug | True | (7,2,1) |
| 45 | Lug | Lug | True | (9,0,1) |
| 46 | Lug | Lug | True | (9,0,1) |
| 47 | Lug | Lug | True | (8,0,2) |
| 48 | Lug | Lug | True | (10,0,0) |
| 49 | Lug | Lug | True | (6,2,2) |

| | | | | |
|---|---|---|---|---|
| **50** | **Lug** | **Lug** | **True** | **(7,2,1)** |

Table 5-3: Results for Classification

The probability with which the object is correctly classified is .71±.07.

7% is the ambiguity that is found while testing.

## 5.4 System performance

| **Data Set** | | **Time at which lug actually abandoned** | **Abandoned luggage Detected** | **Alarm Generated** |
|---|---|---|---|---|
| **Pets2006** | **Video1** | **1:33** | **1:44** | **2:07** |
| | **Video2** | **1:08** | **1:22** | **1:42** |
| **iLids_AVSS** | | **00:38** | **00:41** | **00:50** |
| **Pets2007** | | **Never** | **False warning generated** | **False alarms generated** |

Table 5-4: Results for overall system

Results and evaluation shows that the system generates correct warnings and alarms with 75% accuracy.

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

Our system detects abandoned luggage in surveillance environment, through foreground mask sampling, only the object of interest is localized, while filtering out all irrelevant , interfering agents. Haar classifiers and sift features are used to identify humans , luggage and trolleys in the scene. These techniques improve accuracy of abandoned luggage detection.

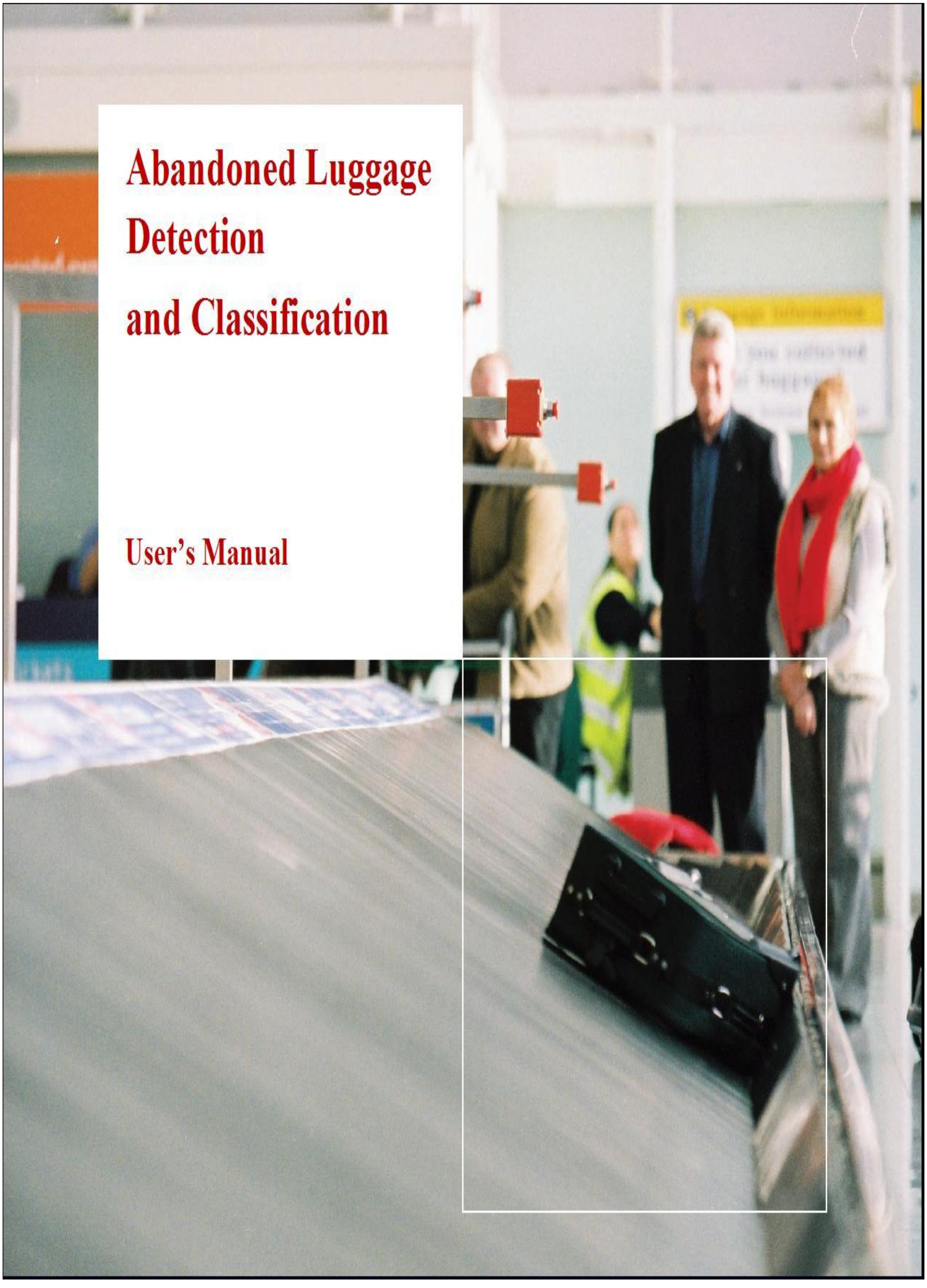
With the advancement in visual and programming technologies, many improvements and additions can be made to the system. In future the ISC system can be extended to real time application. Moreover it can be extended to multi camera network in which coordination of various cameras enables cues to gathered from multiple perspectives and information to be relayed from one another camera. High population density is a difficult issue for a vision based methods , even humans can not notice abandonment reliably. So an object recognition based solution can be added to detect the abundant luggage. More classes like group of humans can be added to the object classification module.

# APPENDIX A-1

# USER MANUAL

# Abandoned Luggage Detection and Classification

## User's Manual



# TABLE OF CONTENTS

# 1. How To Use Your Manual

This manual is your guide to the system Abandoned luggage detection and classification. It contains essential instructions for setup and operations.

The system has an interface which allows you to directly interact and monitor the automated system. Make sure that you are familiar with its operation before installing in the environment.

# 2. Installation

The product is a software with easy to install modules. It is an automated system and has been developed in OpenCV . System will have an interface for end users, who will use the system in their organization. Interface is user friendly in all respects.

# 3. Features

Interface of the system has three buttons

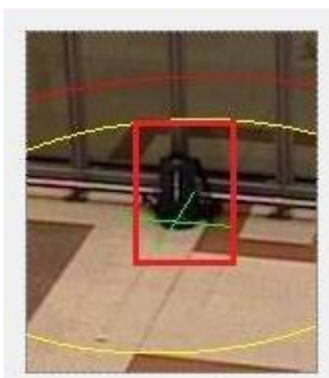
1. Start Surveillance
2. Stop Surveillance
3. Details

Interface has two windows to show live video from the surveillance cameras. Proper error messages will be displayed in case of in faults or exceptions. Figure 1 shows the interface.


Figure 1: System interface


This button allows you to start the surveillance. When pressed, system starts capturing and processing video.


You can stop the surveillance by using this button. Make sure that you do not halt the processing and press this button at the end.



This window displays the luggage detected and classified. Yellow circle shows safe zone and warning is generated when owner is out of red circle.

This window differentiates the foreground from background for better understanding and visualization.

# 4. Troubleshooting

If you experience a problem with your system, do not worry. System has a simple design that makes troubleshooting easy and quick, even if you know very little about the software. More than likely you will be able to solve the problem after only few minutes.

## 4.1 Main Cause of problems

**Problem when stopping the surveillance**.

When the surveillance is stopped while system is processing, it halts and cannot be resumed. You will have to start capturing again. To recover from this state, press ESC.

# References

[1] Álvaro Bayona, Juan C. SanMiguel, José M. Martínez. “Stationary Foreground Detection using Background Subtraction and Temporal Difference in Video Surveillance”

[2] Jing-Ying Chang, Huei-Hung Liao, and Liang-Gee Chen. “Localized Detection of Abandoned Luggage”

[3] Geoffrey Samuel, Dr. Honghai Liu. “Comparison of complex-background subtraction algorithms using a fixed camera”

[4] Chris Stauffer , W.E.L. Grimson. “Adaptive Background Mixture Models for Real-Time Tracking”

[5] Xuli Li, Chao Zhang, Duo Zhang. “Abandoned Objects Detection Using Double Illumination Invariant Foreground Masks”

[6] K. Sandeep and A.N. Rajagopalan. “Human Face Detection in Cluttered Color Images Using Skin Color and Edge Information”

[7] Keni Bernardin, Alexander Elbs, Rainer Stiefelhagen. “Detection-Assisted Initialization, Adaptation and Fusion of Body Region Trackers for Robust Multiperson Tracking”

[8] Bo Wu, Ram Nevatia. “Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors”

[9] Kevin Smith, Pedro Quelhas, and Daniel Gatica-Perez. “Detecting Abandoned Luggage Items in a Public Space”

[10] Zoran Zivkovic and Ben Kr¨ose. “Part based people detection using 2D range data and images”

[11] Toh koh ling, Wing teng ho, Chee wei lee, Yong haur tay. ”Human detection using motion feature extraction and adaptive boosting”

[12] Irshad Ali and Matthew N. Dailey. “Multiple Human Tracking in High-Density Crowds”

[13] Edouard Auvinet, Etienne Grossmann, Caroline Rougier, Mohamed Dahmane and Jean Meunier. “Left-luggage detection using homographies and simple heuristics”

[14]Medha Bhargava,Chia-Chih,M.S.Ryoo and J.K.Aggarwal . “Detection of Abandoned Objects in Crowded Environments”

[15] Biswajit Bose. “Classifying Tracked Objects in Far-Field Video Surveillance”

[16] Mitesh Gupta, Shishir Jain. “Detection of Left Luggage and Theft”

[17] Diedrick Marius, Sumita Pennathur, and Klint Rose. “Face Detection Using Color Thresholding, and Eigenimage Template Matching”

[18] Sancar Adali,Hasan Ayaz,Ali Oğuz Ūstūn. “Tracking human faces using motion and background subtraction”

[19] Ying-li Tian, Rogerio Feris, Arun Hampapur. “Real-Time Detection of Abandoned and Removed Objects in Complex Environments”

[20] Richard A. Halliwell . “Real time full body gesture recognition system”

[21] Prithviraj Banerjee and Somnath Sengupta. “Human Motion Detection and Tracking for Video Surveillance”

[22] Vincent Urias, Curtis Hash . “Detection of Humans in Images Using Skin-tone Analysis and Face Detection”

[23] Radu Bogdan Rusu, Andreas Holzbach, Michael Beetz. “Detecting and Segmenting Objects for Mobile Manipulation”

[24] Birgi Tamersoy. “Background Subtraction”

[25] Ying-Li Tian, Max Lu, and Arun Hampapur. “Robust and Efficient Foreground Analysis for Real-time Video Surveillance”

[26] T. Bouwmans, F. El Baf, B. Vachon. “Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey”

[27] Zoran Zivkovic, Ferdinand van der Heijden. “Efficient adaptive density estimation per image pixel for the task of background subtraction”

[28] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”

[29] Nor Amizam Jusoh and Jasni Mohamad Zain. “Application of Freeman Chain Codes: An Alternative Recognition Technique for Malaysian Car Plates”

[30]O. Javed and M. Shah, “Tracking and object classification for automated surveillance”

[31] Ahmed Fawzi Otoom, Hatice Gunes, Massimo Piccardi. “Automatic Classification of Abandoned Objects for Surveillance of Public Premises”

[32] L.M. Brown, “View independent vehicle/person classification”