Surveillance System Using Abandoned Luggage Detection

iBlobU

Delft, 4th July 2007

Man-Machine Interaction Group
Faculty of Electrical Engineering, Mathematics and
Computer Science
Delft University of Technology, the Netherlands

TU Delft
Technische Universiteit Delft

Surveillance System Using Abandoned Luggage Detection

Bachelor Thesis (IN3700)
in Media & Knowledge Engineering

Man-Machine Interaction Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
The Netherlands

4th July 2007

Man-Machine Interaction Group

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

Members of the Assessment Committee

dr. drs. L.J.M. Rothkrantz
ir. M. Sepers
ir. Z. Yang

Members of iBlobU Team

Calvin Choy	(1230514)
May Chung	(1174428)
Gherry Harahap	(1217461)
Meriane Natadarma	(1203819)
Sin Lin Wu	(1169912)

Surveillance System Using Abandoned Luggage Detection

Calvin Choy, May Chung, Gherry Harahap, Meriane Natadarma, Sin Lin Wu
Man-Machine Interaction Group
Faculty of EEMCS
Delft University of Technology

Members of Assessment Committee

dr. drs. L.J.M. Rothkrantz, ir. M. Sepers, ir. Z. Yang

Abstract

Many public or open areas are facilitated with cameras at multiple angles to monitor the security of that area for keeping citizens safe. This is known as the surveillance system. At the moment, the best solution to reach a safe environment requires a human operator tools to monitor the digital camera images. Even though human is arguably the most intelligent creatures in this world, there are still some shortcomings in the existing solution. In addition, the reason increased adaptation of security cameras to survey public areas has caused a digital data explosion that human operators have difficulty to keep up with. Because of these shortcomings, humans keep inventing new discoveries to make the best of it. In order to support this surveillance system, a recognition and tracking system is built in this project to detect an abandoned luggage in the public transportation area such as train stations and airports. The goal of this project is to design and implement a system which will be able to detect abandoned luggage using the captured images or videos from the camera as the input of the system. The system realizes image segmentation and image tracking, creates blobs of objects, labels the blobs and finally gives a warning when an abandoned luggage is detected. Also a database is developed to store all the media data.

Acknowledgement

This bachelor's thesis describes the research and development that was done to accomplish the bachelor project. The project was carried out at the Man-Machine Interaction Group, of the faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology.

First of all, we would like to thank our thesis supervisor, Drs. Dr. L.J.M. Rothkrantz for his guidance and support. His knowledge and ideas have given us a lot of inspiration.

Secondly, we would like to thank our advisor, Zhenke Yang, for his ideas and suggestions, Dragos Datcu for giving us information about the OpenCV library and also Toos Brussee-Donders for arranging the logistic during the movie shoots at Delft-Zuid Train Station.

The visit to NS Amsterdam and NS Utrecht gave us much information about the security related work and the existing systems. Therefore, we want to thank Jan Dietz, Monique Holland, Cees Roffel, Mark van Teeffelen, Martin de Vries and Hans Heida for their time and kindness.

A big thanks to all of our friends and family who helped and supported us (in)directly with the project. Their help and support motivated us to finalize this project.

The iBlobU Team
Calvin Choy
May Chung
Gherry Harahap
Meriane Natadarma
Sin Lin Wu

Table of Contents

Appendices

List of Figures

List of Tables

1 Introduction

"The concept of progress acts as a protective mechanism to shield us from the terrors of the future." Frank Herbert

After the attacks of 11th September 2001 with the airplanes at the Twin Towers in New York and the Pentagon in Washington DC, the fear of terrorism has grown amongst people in the world. There were threats for more attacks and the world lived in fear. Then on 11th March 2004 there were the attacks in the train in Madrid and on 7th July 2005 the subway stations in London.

Figure 1: Terrorist attacks at different places around the world

As a result, people feared to take public transportation with the attacks in their mind. When using public transportation, people now tend to be more scared for abandoned luggage and suspicious behaviour of travellers. In Amsterdam, the whole railway system went down out of precaution when travellers spotted two suspicious men in a train and alarmed the police. To provide people a safe feeling when travelling with public transportation, it is necessary to have better security systems at transportations area and their surroundings. Security cameras that can recognize suspicious circumstances automatically are convenient in this case. Even though security guards are watching the security videos, they are not always able to detect all the crime. With software that is able to automatically detect crime, the guard will be warned and he can watch at the videos and trigger an alarm if necessary.

However, such an automated system will cause discussions in the society because applying such a system publicly can influence the daily life of people. Besides that, the infrastructures that are needed cannot be considered as cheap. This implies that designing such an automated system should be done carefully and that the societal analysis becomes an essential part in the design of the system. As for technology, the possibilities have to be investigated elaborately in order to be able to maximize the result and minimize the costs.

Within this framework, an analysis of a surveillance system which is widely adopted nowadays will be done, since an automated system will support the activities that are carried out in this surveillance system.

1.1 Societal Relevance of Surveillance System

In this section, the surveillance system that is currently used will be analyzed. In this manner, the advantages and disadvantages can be identified. These findings will be used in the design of the automated system, in which the advantages will be preserved and disadvantages will be removed.

1.1.1 Current Situation

Since the attacks of 11th September 2001 in USA and July 2005 in London, an increasing number of places are covered with Closed-Circuit Television (CCTV) cameras to prevent terrorism. Cameras are on public buses, airports and in railway stations, building lobbies, schools and stores. The captured videos are fed to central control rooms with the security staff monitoring the videos. By that, the security staff will be able to detect suspicious activities and thus ensure public safety. For this reason, the use of surveillance cameras has increased rapidly and has become a part of the daily life in public and private places.

The main function of using the cameras is not only for the crime prevention, but also for the crime detection and prosecution. The terrorist who have committed the attacks in London at July 2005 have been found by using the CCTV videos. Cases such as murder or robbery can also be solved by using the CCTV evidence. With the rapid advance of today's technology, cameras are getting 'smarter'. As a result, the cameras will not only watch the places, but will also be able to interpret and analyze the observed objects to detect unusual activities at early stage.

1.1.2 Situation Analysis

Although public areas are observed by many surveillance cameras, humans can monitor a few cameras at a time. This causes the scarcity that humans are unable to observe all kind of situations on certain cameras simultaneously. If a crime is committed, the cameras are only able to help with the investigation. Also the tasks of the security staff includes more than just sitting back on their chair and watch the videos 24 hours a day. They also need to inspect the environment, write logs and solve disturbances. Because the workload is more than just monitoring, it may occur that they miss some captured scenes.

Another problem is that there are many cameras and too few personnel to watch every single camera. In the UK only, with more than a million CCTV (Closed Circuit Television) cameras [3], it is becoming increasingly difficult for the security staff to manage. Besides the lack of personnel, watching the monitors also demands a higher level of concentration than many people can manage. Usually the concentration of the security guards will drop after an amount of time. Another problem is that video monitoring is very subjective. Since everyone has a different description for suspicious, it may happens that one security guard observes a dubious looking activity that another security guard does not even notice or mark as suspicious.

1.1.3 Ethical Issues

As the use of surveillance cameras has increased rapidly, privacy concerns have also raised. With so many cameras all over the place, it is possible that a certain person is videotaped on different cameras. This will give that person a feeling of privacy invasion, which has a negative impact on civil liberties. Questions about who has access to these videos have also risen due to the possibility of abusing these videos by certain people. Therefore, the authorities must ensure that such surveillance system is strictly monitored by authorized personnel. Besides the privacy, trust to the system is also an issue. Trust is concerning the reliability of unattended luggage detection or suspicious behaviour of a person. If the system gives many false alarms, then the cameras certainly need to improve before it can be widely used in order to maintain public safety.

1.2 Problem Definition

After analyzing the current situation and other societal problems, the problem can be defined as:

> *"Develop a user-friendly system which is able to detect abandoned luggage in public transportation and surroundings using video captures as the input of the system. When a suspicious left behind luggage is detected, the system must trigger a warning signal to the user"*

1.3 Concept of Solution

One of the most common problems that security staff faces, is to detect whether people's behaviour is suspicious and whether an abandoned luggage in a busy railway station, for example, is harmless. With the help of 'intelligent security cameras', it is possible to solve this problem. Such cameras will be able to analyze video images and spot potential threats, from abandoned luggage to unusual behaviour of people that might be missed by manual video monitoring. The ability of the cameras to identify exceptions allows personnel to focus on the important events or to provide substantial operational benefits. Those cameras can be programmed to look for a certain object, like abandoned luggage. Still, there are liability issues, such as false positives luggage in correctly classified as abandoned. Therefore, the technology for those cameras needs to be improved before it can be widely used. Nevertheless, humans are still crucial to tell whether an abandoned luggage, tagged as a threat, is truly a danger or just a false alarm. For the subjective view problem, gaining experience on various human behaviours and how to monitor the video effectively are crucial. Another possibility could be by providing workshops where all important cases that need certain care are discussed.

1.4 Project Goals

The broadness of an automated system, such as stated in the Problem Definition, made it necessary to reduce the scope of the system in order to make it more feasible to be developed within two months. Instead of full dedication to design and implement such an automated system, we have chosen also to provide knowledge and resources for further research in this field. This brought us to the main goals in the scope of this BSc project.

1. Doing field research on the current video surveillance systems applied in the train and train stations.
2. Designing and implementing a database system where the information about the contents of the videos and photos are able to be managed.
3. Produces videos and photos that illustrate the behaviour of the passenger with its luggage. These videos and photos can be useful as training data.
4. Designing and implementing an abandoned luggage detection system on the train or the area around it. This system will trigger a warning when an abandoned luggage is detected. This system will, however, have lots of constraints which are enumerated in the System Analysis chapter at section Scope of This Project. The goal is met when the system has been given a video from around the bench in the train station and it triggers a proper warning. See Appendix D for the script of this video.

1.5 Project Activities

During this bachelor project, there were other activities that support the design and implementation of the system. At the beginning of the project, literature survey was conducted and some dummy data was made. The interviews with the current system's users were held during the visit to the surveillance chamber or so called *controle kamer* at the train station in Amsterdam and Utrecht. The results of the interviews can be found in Appendix F and G. Some photo and video shoots were taken at Delft-Zuid Train Station and at TU Delft EWI practical hall. These results will be used in this thesis as a reference of illustration of our implemented system.

1.6 Outline of the Thesis

This thesis is structured as follows. The working process of this project is described in Chapter 2. This includes the planning and the tools used for communication between the team members. In Chapter 3, theoretical backgrounds are provided that are relevant to our project along with summaries of available implementations of the abandoned luggage detection system. In the System Analysis chapter, a detailed analysis of the *iBlobU Detection System* and the *iBlobU Database* is given. It begins with the description of a proposed system for detecting abandoned luggage. Also, it justifies the reason of implementing the *iBlobU Detection System* and the *iBlobU Database* independently. The requirements that both systems have to meet are also briefly discussed in Chapter 4.

For the design of the software which will be described in chapter 5, a global design of the proposed system will be given first, followed by the detailed design of the implemented software. It contains the architecture, model and the UML of the software system. Chapter 6 will take the implementation of the *iBlobU Detection System* and *iBlobU Database* in consideration. The techniques and algorithms which have been applied during the implementation of the applications are explained. In the Evaluation chapter which follows after the implementation chapter, procedures for testing the software results during the implementation process are described. Software and user tests are also described and evaluated. Improvement recommendation is given as a conclusion to this chapter. The last chapter concludes this thesis. The goals which are given in the Introduction are discussed and ideas for future work are provided.

2 Project Management

"True interactivity is not about clicking on icons or downloading files, it's about encouraging communication." Edwin Schlossberg

Communication is important when working in a team, especially when the team consists of many members. Also it is important to make agreements at the start of the project regarding the way of work. Another important thing in a team work is the planning. Without a planning, it would be like sailing on sea without a compass. Every team member knows what the goals of the project are by following this planning as a guideline. It is required that everyone follows the guideline to work efficiently and of course to achieve the goals of this project.

2.1 Planning

For this project, there were few job roles applied to every member of the group. The job roles that were made are the chairman, the secretary and the team members. The chairman holds the briefings, before and after work time, and the secretary documents the process of all meetings. The chairman and secretary are functioning as the contact person to communicate with the supervisors. With these roles, every team member knew to whom he or she had to go when a task was finished and another task was needed. The division of task can be found in table 1.

Table 1: Division of task

	Role	
Week	Chairman	Secretary
1	Calvin	Gherry
2	Gherry	May
3	May	Meriane
4	Meriane	Sin Lin
5	Sin Lin	Calvin
6	Calvin	Gherry
7	Gherry	May
8	May	Meriane
9	Meriane	Sin Lin
10	Sin Lin	Calvin

At the start of this project, a planning was made that covered the entire time span, this is shown in table 2. When comparing this planning with the actual time table, which is shown in table 3, it is noticeable that the actual planning took more time. Instead of the 10 weeks that were planned, the project took 12 weeks to complete.

Table 2: Planned time table

	Week									
Task	1	2	3	4	5	6	7	8	9	10
Proposal	🟩									
Research		🟩	🟩	🟩						
Requirements Document		🟩								
Database Design		🟩	🟩							
System Design		🟩	🟩							
Specification			🟩							
Movie Scripts		🟩								
Movie Shoots			🟩							
Photo Shoots			🟩							
Questionnaire		🟩								
Interview in Amsterdam			🟩							
Interview in Utrecht			🟩							
Database Implementation				🟩	🟩					
System Implementation					🟩	🟩	🟩	🟩		
Evaluation									🟩	
Final Report									🟩	🟩
Presentation Preparation										🟩

One of the reasons for the delay is that the implementation phase took longer than expected. The libraries that are available were not suitable for the system. This problem was solved by building our own library which is also time consuming. The interviews were planned in the first few weeks, unfortunately the appointments for the interviews were so difficult to make. In this case, the interviews could give insight on the procedures when abandoned luggage is found. Because the interviews were in the last few weeks, it was difficult to know what functions were needed.

As said earlier, the actual planning took more time, but the division of roles ends in week 10. The division of roles wasn't necessarily anymore in the last two weeks, because it was clear to everyone what needed to be done and by whom.

Table 3: Actual time table

	Week											
Task	1	2	3	4	5	6	7	8	9	10	11	12
Proposal	🟩	🟩										
Research		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
Requirements Document		🟩	🟩	🟩								
Database Design		🟩										
System Design		🟩	🟩									
Specification				🟩	🟩							
Movie Scripts		🟩										
Movies Shoots			🟩	🟩				🟩				
Photo Shoots				🟩								
Questionnaire		🟩				🟩	🟩	🟩				
Interview in Amsterdam								🟩				
Interview in Utrecht											🟩	
Database Implementation			🟩			🟩	🟩	🟩	🟩			
System Implementation			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	
Evaluation												🟩
Final Report									🟩	🟩	🟩	🟩
Presentation Preparation												🟩

2.2 Teamwork

All team members have worked with each other on other projects of the bachelor curriculum, although never on a fulltime project. To achieve effective teamwork during the project, the following decisions were made regarding the teamwork.

General decisions:

- every morning a briefing
- every afternoon an end briefing
- at the end of the week, the secretary will write a weekly report. In a weekly report all activities of that week are summarized
- every Friday a meeting is held with our supervisor
- all notes of the briefings and meetings are to be put on the Wiki by the secretary.

Language decisions:

- all reports will be written in English, except for the interviews which are written in Dutch
- all comments in the source code will also be written in English.

2.3 Used Tools

For the realization of the project two tools were very important, one for communication and one for source code management.

The tool used for communication was a Wiki. Every team member can change, add or remove text here. The Wiki was used for all communication purposes, absence, planning, division of labour and roles, weekly reports and reports to deliver were put here.

When working in a team it will be inevitable that different persons work on the same file. When this happens and no right tools are used it can be a very laborious job to merge the different versions of the files into one. To avoid this, we used SVN. This tool reduces the previously mentioned problem. Also version numbers are given by SVN to every commit. The advantage of version numbers is that the team members can see in an instant that changes have been made. Also when a crucial error has been made by a team member and this was accidentally committed, it is possible to revert to a previous version. When a team member commits a file this person had to write in the log what he or she did.

3 Theory

“In theory, there is no difference between theory and practice;
In practice, there is.” Chuck Reid

In this chapter, the most important results of the literature study which was carried out during this project will be presented. In section 3.1, two variants of background subtraction are summarized. In the next section 3.2, the object classification based on silhouette will be discussed. At last, the preceding work on detecting abandoned luggage will be recapitulated and the techniques that are used will also be provided.

3.1 Background Subtraction

Background subtraction is the technique that is used to separate between foreground pixels and background pixels in an image. This technique is mostly applied in the surveillance system to detect the movement in a video. The essence of this technique is comparing the images with the background image of the same scene, which is taken in advance. In this section, there will be two variants of background subtraction explained:

- Naïve approach of background subtraction
- Background subtraction by codebook subtraction.

3.1.1 Naïve Approach of Background Subtraction

In this variant, the difference of brightness between pixel of the image and the background image is calculated for all pixels in the image. It is assumed that there are no foreground object(s) in the background image.

To obtain the brightness of the image, the colour images must be converted into greyscale images. In a greyscale image, one pixel contains only one value that represents the brightness of the corresponding pixel. In the case of an image with RGB colour channel, the conversion to a greyscale image can be done by calculating the value of the brightness by means of the following formula:

$$0.3 * \text{red} + 0.59 * \text{green} + 0.11 * \text{blue} \qquad \textbf{(Eq. } 3.1)$$

The result of this calculation can be seen in figure 2:

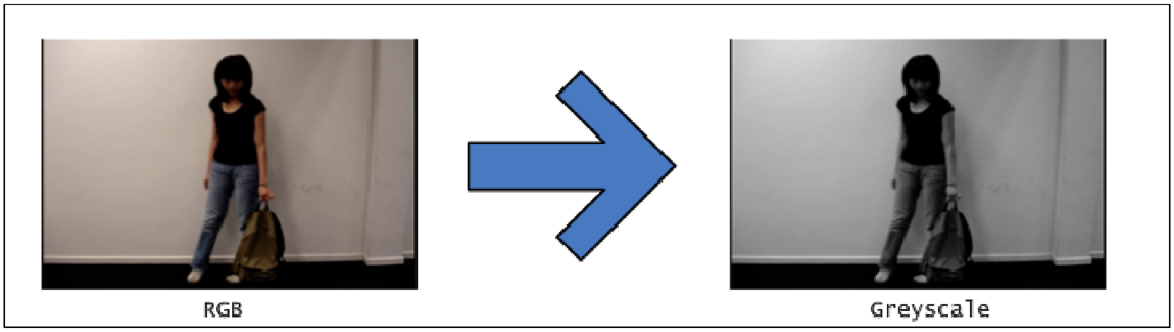


Figure 2: Conversion from RGB colour image to grey picture

The difference between two brightness values is calculated by subtracting the pixel brightness of the foreground image from the pixel brightness of background image and then taking the absolute value of it.

$$Brightness's\ difference = \left| brightness_{image\ with\ foreground} - brightness_{background\ image} \right| \quad \textbf{(Eq.}\ 3.2)$$

When the difference is already calculated, the foreground and background can be defined by setting a threshold value that will distinguish whether a pixel belongs to a foreground pixel or background pixel. This threshold value is pre-defined and obtained after testing all possible threshold values manually. If the difference value of a pixel is larger than the threshold value, then the corresponding pixel can be classified as foreground, otherwise it will be classified as a background pixel. The process of difference calculating and thresholding is depicted in figure 3. White pixel in the thresholded image represents foreground pixel and the black pixel represents background pixel.

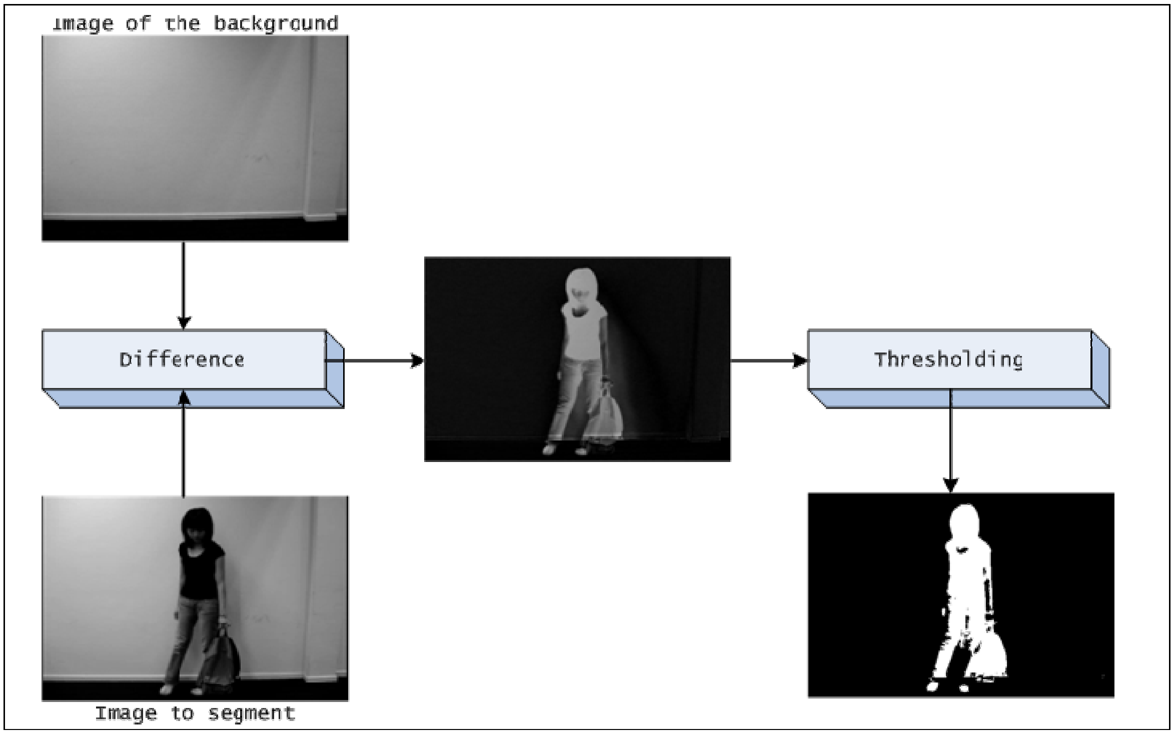


Figure 3: Calculating the difference between background and current frame image and applying the thresholding

Since the result of the segmentation is a binary image, morphological operations such as dilation and erosion can be applied to refine the edge of the segmented object.

3.1.2 Background Subtraction by Codebook Construction

This technique is presented in [4] and there are two phases in this variant:

- Modelling
- Detection

In contrast to the naïve approach of background subtraction, the colour information is also used to segment the foreground in this variant of background subtraction.

3.1.2.1 Modelling

Before performing the background-foreground segmentation, a background model has to be built. The background model of a pixel will be encoded in a codebook, with a multi-dimensional array as a result when a background model of an image is built. The algorithm, which is applied to this technique for background modelling, adopts a clustering technique, inspired by Kohonen, to construct a background model from long observation sequences [5].

For each codebook, it will contain codeword(s) and a codeword is defined as a 6-tuple and a RGB vector. The 6-tuple will hold these variables:

- $\check{I}$ = the min brightness of all pixels assigned to this codeword
- $\hat{I}$ = the max brightness of all pixels assigned to this codeword
- f = the frequency with which the codeword has occurred
- λ = the maximum negative run length (MNRL) defined as the longest interval during the training period that the codeword has NOT recurred
- p = the first access time in which the codeword has occurred
- q = the last access time in which codeword has occurred

And the RGB vector $v_i = \left(\bar{R}_i, \bar{G}_i, \bar{B}_i\right)$ is a vector where the RGB colour information of the background pixel is stored.

The modelling involves testing the difference between the current image and the background model with respect to colour and brightness difference. To determine the colour difference, the colour distortion is calculated. Given an input pixel $x_t = (R, G, B)$ and a codeword c_i where

$$v_i = \left(\bar{R}_i, \bar{G}_i, \bar{B}_i\right) \tag{\textbf{Eq.} 3.3}$$

$$\|x_t\|^2 = R^2 + G^2 + B^2 \tag{\textbf{Eq.} 3.4}$$

$$\|v_i\|^2 = \bar{R}_i^2 + \bar{G}_i^2 + \bar{B}_i^2 \tag{\textbf{Eq.} 3.5}$$

$$\langle x_t, v_i \rangle^2 = \left(\bar{R}_i R + \bar{G}_i G + \bar{B}_i B\right)^2 \tag{\textbf{Eq.} 3.6}$$

the colour distortion δ can be calculated by:

$$p^2 = \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2} \qquad \textbf{(Eq.} \ 3.5)$$

$$colordist(x_t, v_i) = \delta = \sqrt{\|x_t\|^2 - p^2} \qquad \textbf{(Eq.} \ 3.6)$$

In this technique, a brightness change is allowed as long as it is still inside the tolerated range. This range $[I_{low}, I_{hi}]$ is defined as:

$$I_{low} = \alpha \hat{I} \qquad \textbf{(Eq.} \ 3.7)$$

$$I_{hi} = \min\left(\beta \hat{I}, \frac{\check{I}}{\alpha} \right) \qquad \textbf{(Eq.} \ 3.8)$$

where α < 1 and β > 1. Experiments show that the value of α is between 0.4 and 0.7 and β is between 1.1 and 1.5. Testing whether the brightness change is between the tolerated range is done by:

$$brightness\left(I, \langle \check{I}, \hat{I} \rangle\right) = \begin{cases} true & \text{if } I_{low} \le \|x_t\| \le I_{hi} \\ false & otherwise \end{cases} \qquad \textbf{(Eq.} \ 3.9)$$

The construction of the codebook to model a background of a pixel location will go as the following algorithm:

I. $L \leftarrow 0, £ \leftarrow \varnothing$ with

- *L* is amount of codewords in a codebook,
- £ is the sets of the codewords in the codebook and
- $\leftarrow$ means empty set.

II. For t = 1 to N do (N is an amount of training images)

(i). $x_t = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$

(ii). Find the codeword c_m in $£ = \{c_i \mid 1 \le i \le L\}$ matching to x_t based on two conditions (a) and (b)

(a) $colordist(x_t, v_m) \le \varepsilon_1$, where ε_1 is the sampling threshold (*colordist* is defined in equation 3.6)

(b) $brightness\left(I, \langle \check{I}_m, \hat{I}_m \rangle\right) = true$ (function of *brightness* is defined in equation 3.9)

(iii). If $£ = \varnothing$ or there is no match, then $L \leftarrow L + 1$.

Create a new codeword c_L by setting

- $v_L \leftarrow (R, G, B)$

- $aux_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$.

(iv). Otherwise, update the matched codeword c_m, consisting of $v_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $aux_m = \langle \check{I}_{\mathrm{m}}, \hat{I}_{\mathrm{m}}, f_m, \lambda_m, p_m, q_m \rangle$, by setting

- $v_m \leftarrow \left(\dfrac{f_m \bar{R}_m + R}{f_m + 1}, \dfrac{f_m \bar{G}_m + G}{f_m + 1}, \dfrac{f_m \bar{B}_m + B}{f_m + 1} \right)$
- $aux_m \leftarrow \left\langle \min\{I, \check{I}_m\}, \max\{I, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \right\rangle$

End for

III. For each codeword c_i, i = 1,...,L, wrap around λ_i by setting

$$\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$$

To reduce the influence of noise during modelling the background model, a temporal filtering step can be run. This can be done by removing the codeword which has a high maximum negative run length (MNRL). Let M and T_M denote the background model and the threshold value, respectively (usually, T_M is set equal to half the number of training frames, N/2), then the temporal filtering can be described with this equation:

$$M = \{ c_m \mid c_m \in £ \wedge \lambda_m \le T_M \} \qquad \textbf{(Eq.}3.10)$$

After the background model is created, the background-foreground segmentation can be done by using this background model.

3.1.2.2 Detection

The subtraction operation $BGS(x)$ of this technique for an incoming pixel value x can be done as follows:

I. $x = (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$

II. For all codewords in M in equation 3.10, find the codeword c_m matching to x based on two conditions:

- $colordist(x, c_m) \le \varepsilon_2$, where ε_2 is the detection threshold
  (*colordist* is defined in equation 3.6)
- $brightness\left(I, \langle \check{I}_m, \hat{I}_m \rangle\right) = true$
  (function of *brightness* is defined in equation 3.9)

III. $BGS(x) = \begin{cases} foreground & \text{if there is no match} \\ background & \text{otherwise} \end{cases}$

.

The process of background subtraction by means of Codebook Construction is depicted in figure 4:

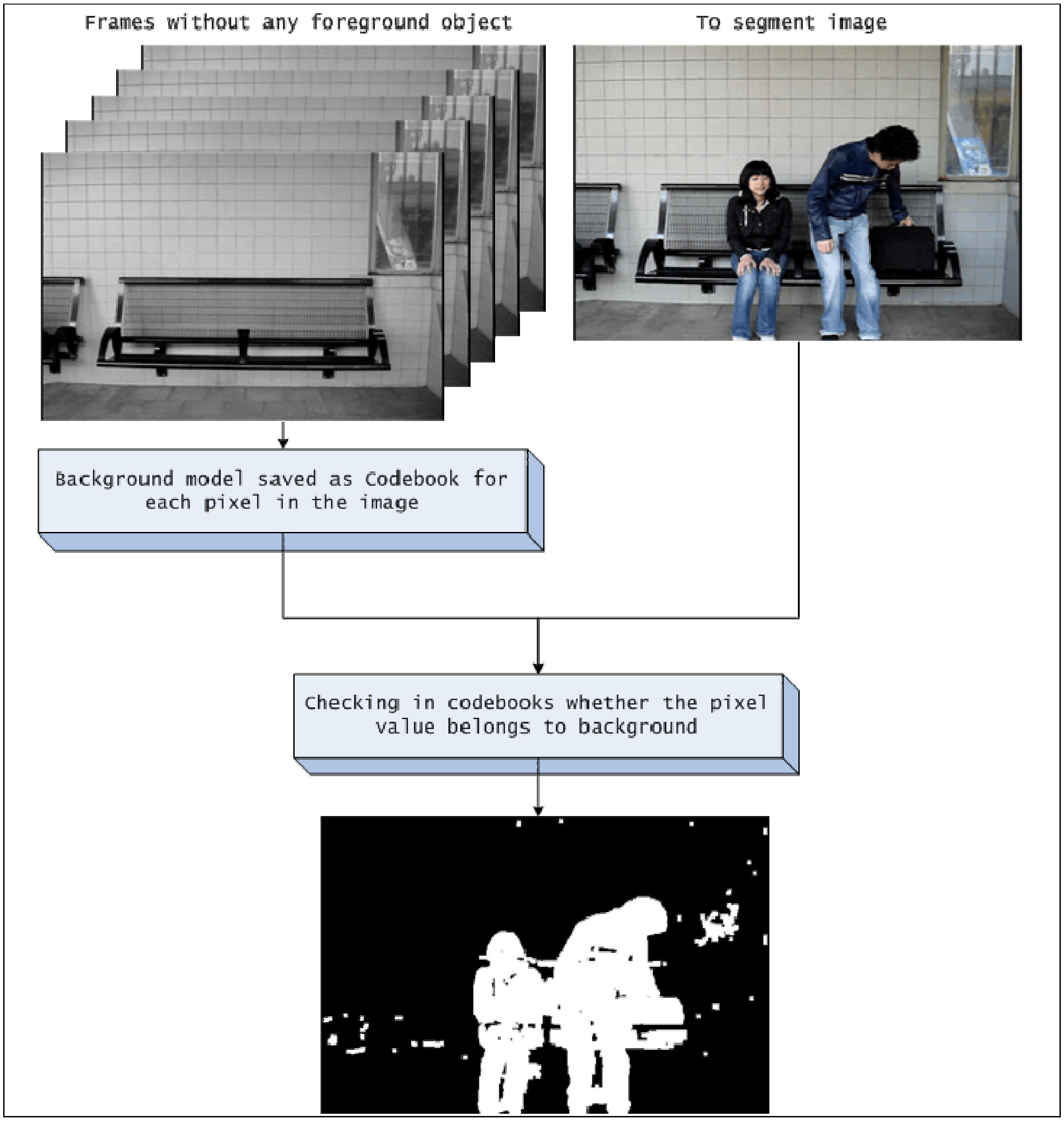

Frames without any foreground object
To segment image
Background model saved as Codebook for each pixel in the image
Checking in codebooks whether the pixel value belongs to background

Figure 4: Background subtraction by Codebook Construction

3.2 Silhouette-Based Object Classification

To understand the behaviour of an object, it is often really important to classify the object first. Classifying an object can be done using features like colour, texture or shape. Silhouette-Based Object Classification is one of the object classifier which analyzes the shape of the silhouette to classify an object. This object classification comes usually handy when an object, which has a typical shape, is to be detected.

In the [1] is explained about the implementation of Object Classification by means of the silhouette. It suggests that given a silhouette as a set of points $\{(x_1, y_1), \mathrm{K}\ , (x_n, y_n)\}$, the centre of mass point $C_m = \left(x_{C_m}, y_{C_m}\right)$ can be calculated as follows:

$$x_{C_m} = \frac{\sum_{i=1}^{n} x_i}{n}, y_{C_m} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad \textbf{(Eq.}3.11)$$

and using the calculated centre of mass the features of this silhouettes can be generated by calculating the distances between the centre of mass point with each point in the silhouette. The Euclidean distance can be used to measure the distance between these two points.

In generating this feature, there are 3 issues to handle:

1. The calculation of the distances has to be done in clockwise direction with the top point of the silhouette as the beginning. This will cause the distance signals of two same object look similar because it is ordered in the same way.
2. The same object can have varying sizes. In order to compare signals corresponding to different sized objects and to make the comparison metric scale-invariant, the distances have to be sampled. Let N be the size of distance signal DS, which has been calculated, and let C be the constant for fixed signal length. The fixed-sized distance signal $\tilde{DS}$ is then calculated by over-sampling or under-sampling the original signal DS as follows:

$$\tilde{DS}[i] = DS\left[i \times \frac{N}{C}\right], \forall i \in \left[1\mathrm{K}\ C\right] \qquad \textbf{(Eq.}3.12)$$

3. To make sure that the distances have the same unit of measure, the distance has to be normalized. The normalization can be calculated with the following equation:

$$\overline{DS}[i] = \frac{\tilde{DS}[i]}{\sum_{1}^{n} \tilde{DS}[i]} \qquad \textbf{(Eq.}3.13)$$

Figure 5 depicts a silhouette and its centre of mass point which is used to calculate the distance with points in the silhouette. Next, figure 6 represents the plot of distance signal of a silhouette.

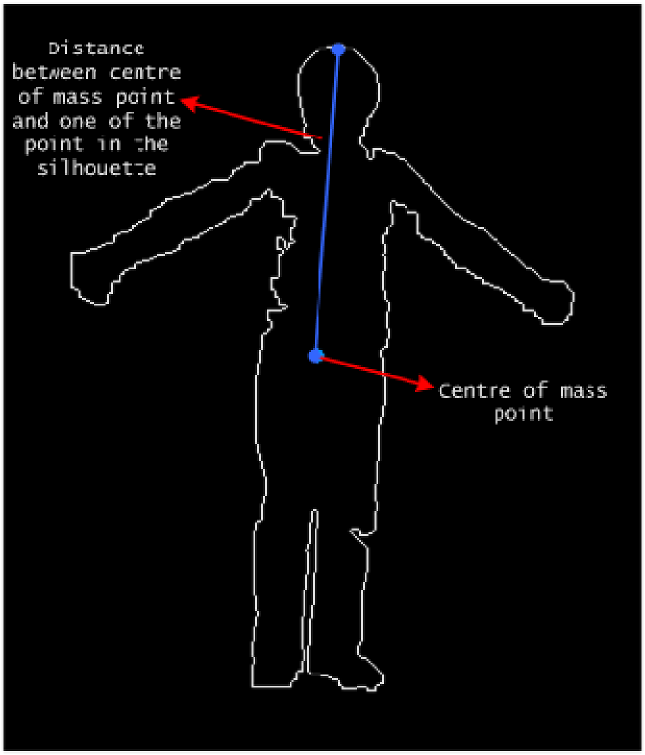

Distance between centre of mass point and one of the point in the silhouette
Centre of mass point

Figure 5: A silhouette with its centre of mass and one of the distances between silhouette points and centre of mass point

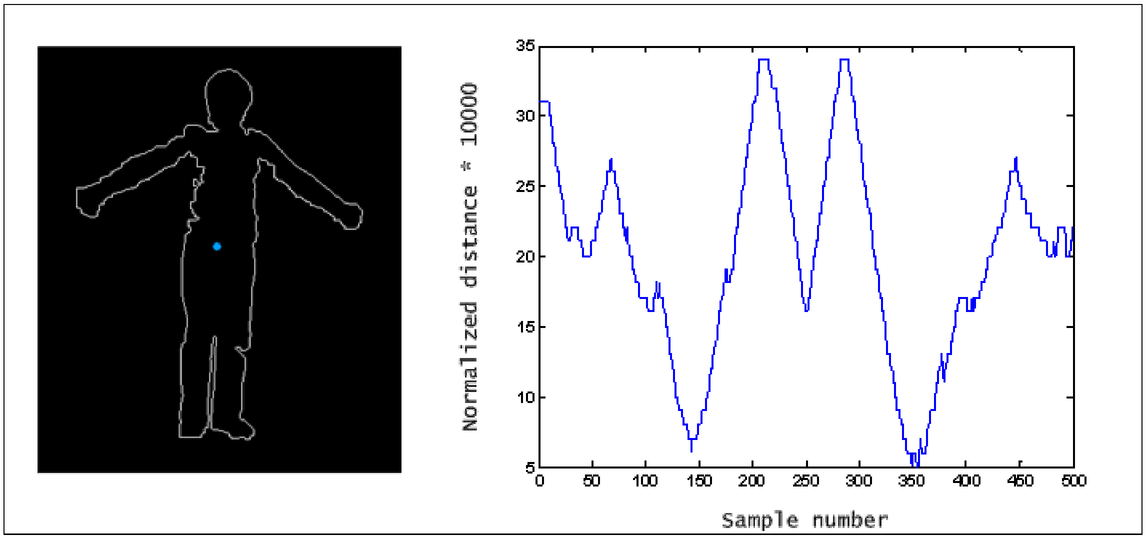


Figure 6: A silhouette and its plot of distance signal

Classifying an object using this distance signal feature can be accomplished by calculating the dissimilarities between the distance signal with the templates of the distance signal which are manually classified beforehand; then, classify it as the object in the template which has the least dissimilarity. The dissimilarity between two distance signal features can be calculated with the following equation:

$$Dissimilarity_{AB} = \sum_{i=1}^{n} \left| \overline{DS_A}[i] - \overline{DS_B}[i] \right| \qquad \textbf{(Eq.}3.14)$$

By setting a threshold value for dissimilarity, this classifier can decide whether the object does not belong to one of the objects in the templates and classify it as an unknown object.

3.3 Related Work

This section will summarize the related work on abandoned luggage detection system. The section begins with brief information about workshop on Performance Evaluation of Tracking and Surveillance (PETS). Next two abandoned detection algorithm which are presented at this workshop will be summarized.

3.3.1 Workshop on Performance Evaluation of Tracking and Surveillance

In recent years, the use of visual surveillance has increased rapidly, as described in Introduction. This is also a major research area in computer vision. The purpose of using visual surveillance is to prevent crime and terrorism, secure public safety and manage transport network and public facilities efficiently. As a result, the necessities for ‘smart’ surveillance system are needed. Such system must be able to observe and obtain detailed information about the activities and behaviours of people. The system will be able to allow security services to respond quickly to potentially critical situations, improving the safety and security in public environments. Therefore, object detection and tracking is very important and challenging, since, we must deal with merging, splitting, entering, leaving and correspondence in crowded areas. The problem becomes more complex, when the public area is observed by multiple cameras at once. Several ‘smart’ cameras have been developed, but they need to be evaluated in order to prove the quality of the implemented system and to highlight the possibilities for system improvement. To achieve this, it is convenient for researchers to use the same video data for evaluation.

Table 4: Brief description of the PETS 2006 datasets

Sequence	Length (s)	Luggage Items	Number people nearby	Abandoned yes/no?	Complexity (rated by PETS)
S1	121	1 rucksack	1	Yes	1/5
S2	102	1 suitcase	2	Yes	3/5
S3	94	1 briefcase	1	No	1/5
S4	122	1 suitcase	2	Yes	4/5
S5	136	1 ski equipment	1	Yes	2/5
S6	112	1 rucksack	2	Yes	3/5
S7	136	1 suitcase	6	Yes	5/5

Therefore, the Performance Evaluation of Tracking and Surveillance (PETS) program was introduced for this purpose. It was first announced in March, 2000. Up until now, nine yearly workshops have been organized to explore various surveillance data, two in conjunction with the IEEE International Workshop on Visual Surveillance. The main focus of this program was to evaluate surveillance algorithms and visual object tracking. Later on, the accent was rested on standard metrics for computing the result of evaluation process. Finally, the most recent is the PETS Metrics which is a newer initiative to provide an online service for automatically evaluating surveillance results. With this, PETS allows researchers to submit their results of object detection and tracking on the web and let the PETS Metrics generate an evaluation of their results.

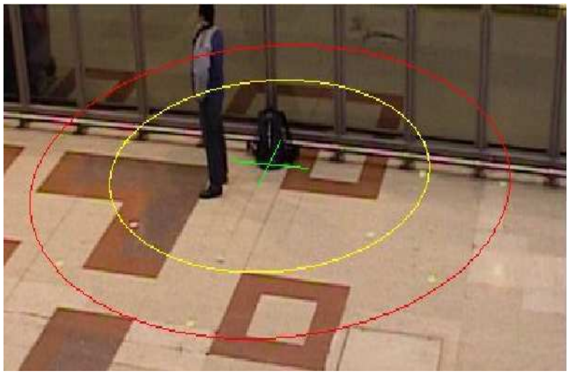


Figure 7: A person within the 2 meter radius (marked in yellow) around their luggage (marked by a green cross)

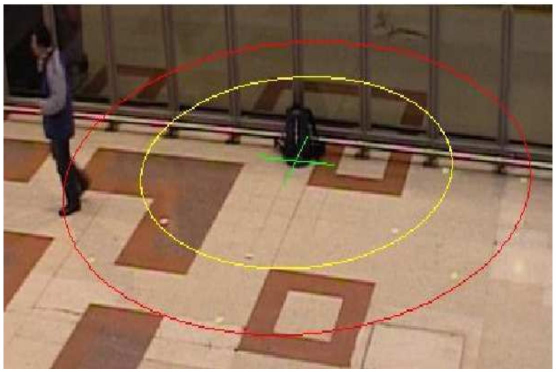


Figure 8: A person crossing the 3 meter radius (marked in red) around their luggage (marked by a green cross)

The most recent workshop is the PETS 2006. The PETS 2006 video data contains seven abandoned luggage scenarios of varying difficulties in real world environment [13][12]. A description of the scenes is given in table 1. Each scenario is filmed from multiple cameras. Those scenes represent a typical security problem, namely detecting abandoned luggage in a busy train station. Ground-truth information such as luggage locations and abandoned-luggage detection times are provided in each sequence. Here, the challenge is to recognize these events, to trigger a warning when it is unattended and to set an alarm when it is abandoned. A luggage is unattended when the owner is within two meter radius of the luggage. When the owner is more than three meters away from the luggage, it is marked as abandoned luggage. These are shown in figure 7 and figure 8.

3.3.2 Used Algorithms at Workshop PETS 2006

Researchers from the PETS 2006 workshop have evaluated the result of their existing abandoned luggage algorithm on the PETS 2006 video. However, only two tracking algorithms will be discussed in this section. But before going into the summaries, the basic steps for developing ‘smart’ surveillance system will be given first.

3.3.2.1 Basic Steps of Surveillance System Development

Most surveillance system uses the following steps to develop a good quality system:

- Object Detection
- Object Tracking
- Object Classification
- Object Recognition
- Alarm Warning

To track the various objects in the scene precisely, the evaluation of *Object Detection* algorithms will have a great influence in the overall performance of the whole system. So, the segmentation between foreground and background object is crucial in *Object Detection*. Therefore, *Object Detection* is normally the first step in the developing process. Tracking the detected object (*Object Tracking*) is for understanding the event, which is observed by the surveillance cameras. The challenge in this step lies in tracking multiple targets which overlap each other within crowded areas. Having the result of the tracking, the object will be classified (*Object Classification*) according to various features such as colour and shape in order to recognize the luggage and persons (*Object Recognition*). After developing all these steps, an *Alarm Warning* will only be given if the detected and recognized luggage is abandoned.

3.3.2.2 Left-Luggage Detection using Bayesian Inference

The system which is presented in this paper for detecting abandoned luggage uses a low-level tracking module and a high-level event inference to solve the tracking and event recognition problem [9]. The tracking module is a combination of a blob tracker and a human tracker, whereas a Bayesian inference framework is used for the event recognition. An overview of this system can be seen in figure 9.

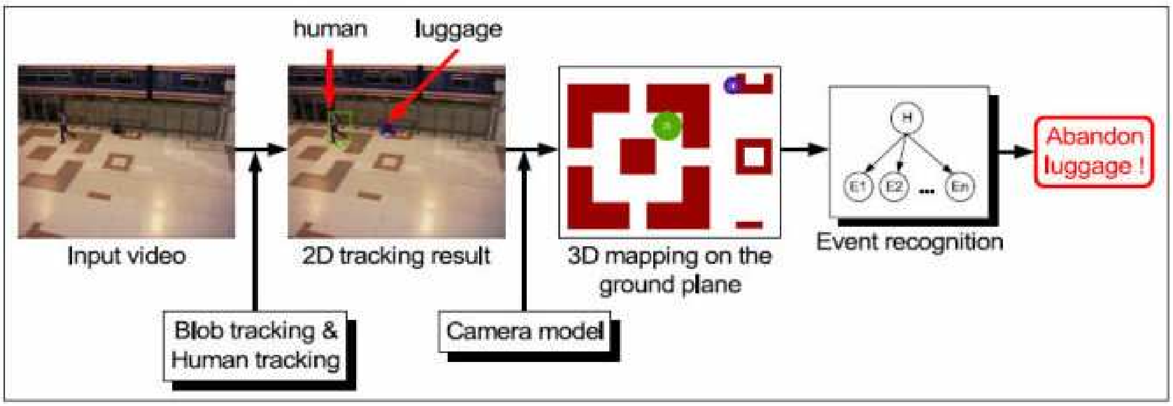


Figure 9: Overview of the system

Looking at the figure, the first step is to detect and track object of interest. The purpose of the blob tracker is to detect and locate luggage. To detect the foreground objects including human and carried objects, a Kalman filter is used [8]. By applying the background model, which is trained using the first five hundreds frame, the pixels of interest is detected and constructed into blobs. Each blob has information about its size, location and appearance (colour histogram). Blob merging and splitting is done by assigning an association value based on overlap between the blob and object bounding boxes, as stated in [9]. Once the blobs are created and all the blob merging and splitting are done, the luggage is detected and located based on their mobility. The reason of using mobility is because luggage hardly moves after it has been tracked. However, this blob tracker has some shortcomings. That is, the tracker can not separate a previously merged human object well. Also, detecting object based on mobility will sometimes lead to misidentifying. The human tracker algorithm based on human shapes, is a simplified version of Wu and Nevada's work [14] and it is only meant for tracking humans. It uses three trained full-body detector, each with a different view (left, front/rear and right), to scan the image. The angles of the detectors are in the range [0°, 45°].The union of the scanning results will be the detected multi-view human detection result. Then, the detected human objects are tracked in 2D using data (frame detection result) association style method. Although the learned detectors are in the range of [0°, 45°], when the angle is a bit to large, detecting human will become less reliable. By combining these two trackers, both limitations will be compensated.

Since the properties of the result of the tracking module are measured in world coordinates, 2D object must be mapped to 3D world coordinates first. This mapping is based on the camera model. Next for the event recognition, an event model for each possible event is developed. Because it is necessary to handle each ambiguity in the event definition, therefore the event model is based on Bayesian inference. Events are considered as hypotheses, whereas related cues as evidences. By detecting abandoned luggage based on those events, an alarm or warning is triggered with high probabilities.

3.3.2.3 Abandoned Object Detection in Crowded Places

To detect abandoned luggage in crowded places when there are multiple people around the area, the method presented in this paper uses two major components [2]. First, there is a *Moving Object Tracking* for tracking and detecting moving objects. Thereafter, a *Stationary Object Detector* is used to mark and detect the abandoned objects. An overview of this method is illustrated in figure 10.

Each detected object is assigned to a unique ID number. This number will be staying with the object during the whole process, from splitting to merging. By using motion, position and region based correspondence, the detected objects are tracked and analyzed from frame to frame to detect object splits and identify the drop-off event (*Drop-off Event Detection*). To identify a drop-off event, the sizes and the motions of the detected child object are inspected and the distance to their parent is also calculated. To calculate the distance, the centre of mass of each object is projected on the ground plan. Then, the projected position will be used to calculate the distance between detected objects.

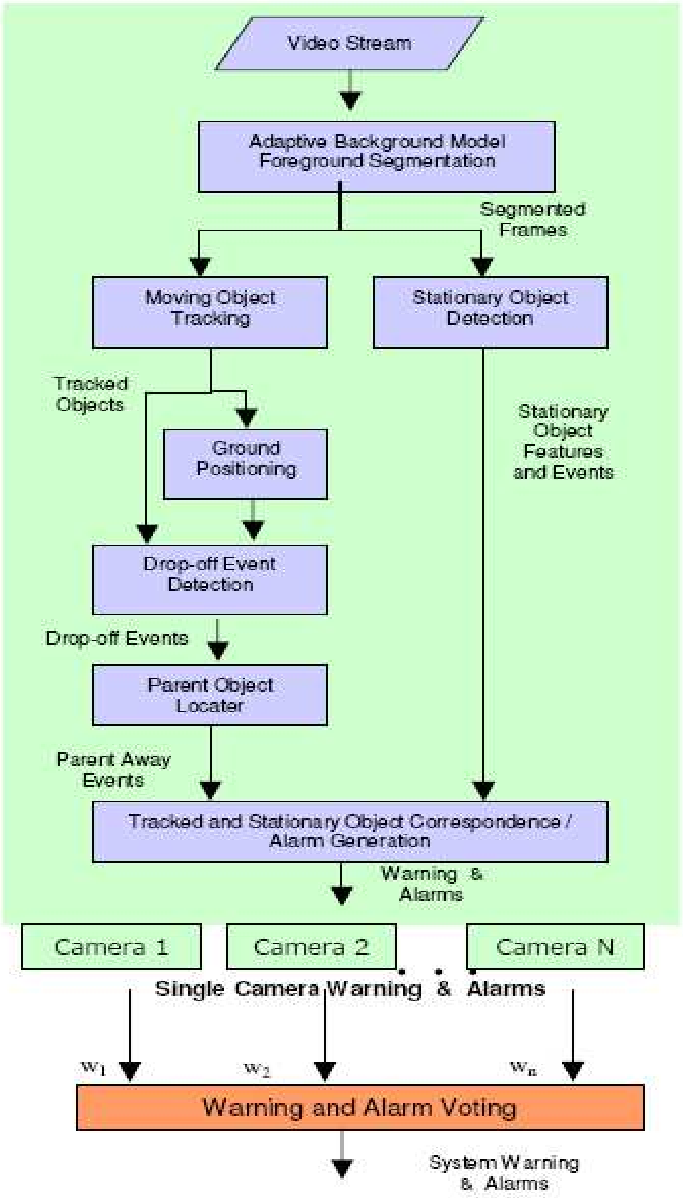


Figure 10: Abandoned object detection method

All the detected child objects are correlated with stationary detected objects to test whether it is truly abandoned. Here, the stationary detected objects are defined as the potential stationary foreground objects such as benches. It uses background statistics with a duration parameter to persistently detect the stationary object. By obtaining all the final abandoned object detection result from multiple cameras and fusing it, warning and alarm is triggered based on the distance between the parent and the child.

4 System Analysis

“Truth is subject to too much analysis.”
Frank Herbert

By analyzing the current situation, it is noticeable that surveillance plays a very important role in the world. Using video surveillance cameras, criminals have been arrested and terrorist attacks have been avoided. Therefore, video surveillance cameras have to be innovated in order to prevent crime and terrorism successfully. An ideal video surveillance system for abandoned luggage detection must be able to detect unusual events, to distinguish between a person and a luggage. When the system has the ability to do this, the behaviour of the person along with his or her luggage can be tracked and analyzed. This is done by comparing real data in normal situation and suspicious situation. By means of this kind of systems, the public transportation provider and the authority have the capability to secure the public safety and therefore, it will give the travellers a saver feeling at the public transportation area.

In this chapter, an analysis of an automated abandoned luggage surveillance system will be given. In the first section, a brief description about the proposed system along with the database system will be given; later on, we will discussed the scope of this project followed by the requirements of the system and the database . An analysis of the environment will be given in the last section.

4.1 The Proposed System

A shortcoming of the manual video surveillance system in the public transportation area is that the security staff cannot observe all suspicious behaviour captured by the video cameras and monitored on the computer screen. As a result, manual system needs to be replaced by an automated system which will alarm the security staff when abandoned luggage is detected or warn the traveller when he or she leaves the luggage unattended. This proposed system aims at detecting abandoned luggage at the train station and it’s surrounding. It will be constructed in three different User Interface versions for three different kinds of users.

Developer:

The first version is a developer version with a Graphical User Interface, only meant for researchers. In this interface, the users will be able to see the output of each processing steps, carried out by the abandoned luggage detection system. The purpose of this interface is to help researcher to analyze en evaluate the quality and performance of the system visually in order to improve the system in nearby future development.

Security Guard:

The second version is a restricted developer version, named security guard version. Generally, the users of this version are the security staffs. Unlike the developer version, the user of this version will only see the end visual output of the process and its related information which is relevant for the users. With this, the users can observe the danger of detected abandoned luggage and therefore take the necessary action.

Embedded camera:

In the third version, the system will not provide a GUI. In this version of the system, the software will be embedded into the camera. These cameras can be employed in public transportation area for warning the travellers when they forgot to bring their luggage after leaving the corresponding area.

In the case with cameras where the software is embedded, the cameras can either work as a stand-alone system or as a network system consisting of multiple cameras that share data. The advantage of using multiple cameras is that the range in which objects can be tracked can be expanded by the number of cameras operating in different location of interest.

In figure 11 below an overview is given for the proposed system

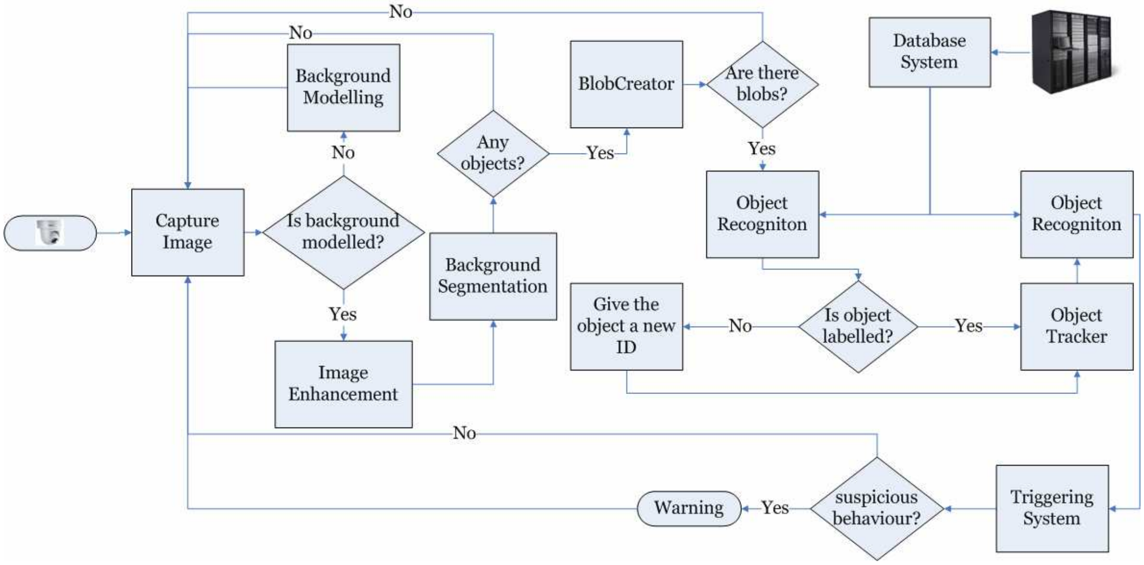


Figure 11: Overview System Flow Diagram

Because the proposed system will be using real-time camera video data, the camera has to be mounted on places where the view is clear and where potential abandoned luggage is likely to be found. Another crucial issue for detecting abandoned objects correctly is light; therefore, the environment where the cameras are mounted should be adequately exposed by the light in order to have least illumination as possible.

As illustrated in figure 11, the proposed system will also use a database. The database will be used by the developer to stored information of gathered media files. The goal of a database is to give developers the possibility to easily store and retrieve data of interest. Depending on the data that the developer wants to use, media files can be retrieved from the database using a user friendly interface. This has the advantage that the developer does not have to go through all the files manually in order to find the media files of interest. Besides the search functionality the database can be used to train the system. Features of sample data can be extracted and used for the classifying process and rules generation when an expert system is used.

4.2 Design Methodology

During the development of the system, an incremental approach will be taken. As described in Chapter 2, Project Management, there are two different prototypes that will be made. Within the time constraint of this project it is important that we can build upon the created prototypes. It is expected that the problem space grows while the development proceed and gets more complex. By creating a simple working prototype as soon as possible, it is possible to evaluate proposed ideas in an early stage and reduce the problem space by detecting problems and difficulties in an early stage. Check points have been set to evaluate the system from time to time while the development proceeds. Since the various parts of the system will be divided among the team members, the checkpoints will be used to evaluate the current state of the overall system. The incremental approach gives the team members the chance to work on the system while exploring the problem space at the same time. After the first two prototypes have been made and evaluated, the team can discuss the problems and difficulties that are left and work towards the final product (see figure 12).

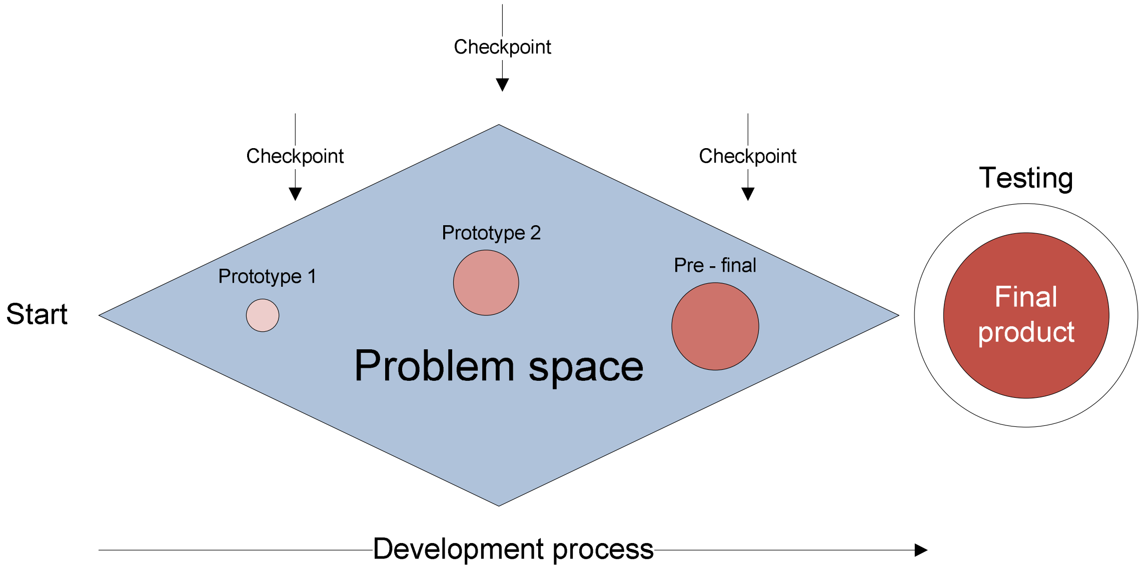


Figure 12: Sketch of the incremental approach for the development of the system

4.3 Scope of This Project

Due to the time constraint of this project, it is not possible to build the whole proposed system. Therefore, the decision is been made to built a restricted version of the proposed system. This system is given the name, *iBlobU Abandoned Luggage Detection System*. Further, for the sake of writing this thesis, we will refer the *iBlobU Detection System* to *the iBlobU Abandoned Luggage Detection System* and *iBlobU Database* to our *Database System*. In the *iBlobU Detection System*, only the developer and security guard version of the proposed system will be implemented. The *iBlobU Database* will also be constructed, but it is not related to *iBlobU Detection System*. It will be used as a stand-alone application for adding, correcting and retrieving media and training data. Thus, the flow diagram of this *iBlobU System* is nearly the same as the proposed system, as depicted in figure 11. The missing parts are the connections between:

- *Database System* and *Object Recognition*
- *Database System* and *Behaviour Recognition.*

This is illustrated in figure 13. The blue region represents the stand-alone database system and the red region represents the *iBlobU Detection System* flow diagram.

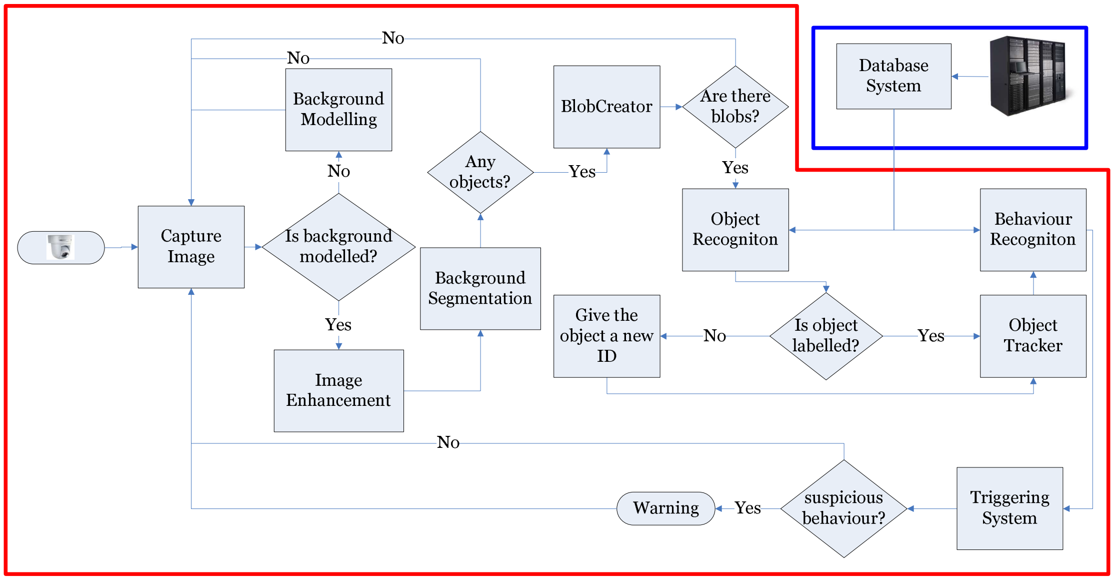


Figure 13: Overview *iBlobU Detection System* flow diagram and *iBlobU Database*

Although the streaming camera will be implemented in this *iBlobU Detection System*, the input will not be a captured image from a streaming camera, but from an AVI video. This AVI video was shot beforehand. The reason for using AVI movie files during the development is that we wanted to test our system using real life environment data where the system is suppose to run. These data were not available and had to be made manually. Other constraints that have been added to the construction of *iBlobU Detection System* are mentioned below.

- *The first hundred frames must be free of moving objects*
  To extract the foreground objects from the background, the background model must be developed first. In order to obtain a decent background model, it needs to be learned from a few numbers of frames. Therefore, the numbers of frames needed to

create a reasonable background model is set to hundred frames, starting from the very beginning and thus must not contain any moving objects such as humans.

- *High contrast between foreground and background*
  Depending on the algorithm used to distinguish the background and foreground, it is important that the contrast is high enough to ensure a clear difference between foreground and background.

- *All foreground objects must be clearly visible*
  The view of the camera must not be blocked and object of interest must be completely visible.

- *Only one luggage will be used*
  Within the scope of this project, only one luggage will be used in scene to test different scenarios.

- *The position and direction of the camera is fixed during the recordings*
  During this project, sample movies will be made using a digital camera. The position and direction of the camera remains fixed during each recording.

- *The light condition must remain constant as much as possible*
  The environment should not be over exposure and light conditions must remain constant as much as possible to avoid occlusion.

- *All frames of the AVI video will be processed*
  This depends on the feasibility of the system to perform all the processing of the *iBlobU Detection system.*

- *The frontal distance between the camera and the moving object must between 2 and 2, 5 metre*
  These are the distances for the movies that are shot, the system will be tested and developed with the movies recorded using these properties.

Besides the above listed constraints, several requirements have to be met by the *iBlobU Detection System*. The *iBlobU Database* which is a stand-alone application must also meet some requirements. All these requirements will be discussed in the following section.

4.4 Requirements Analysis

As mentioned in the previous section during the scope of this project two systems will be implemented as proof of concept for an abandoned luggage detection and one database system will implemented to store information of media files. Although the database system will not be coupled with the system during this project, it will be implemented as a separate product. In the rest of this section functional and non - functional requirements will be specified for each of the systems that are going to be implemented within the scope of this project.

4.4.1 Functional Requirements

The *iBlobU Detection System*

The *iBlobU Detection System* will be used by two types of users, the security staff in the control room and the developer. Although two version of the system will be made to fulfil the needs of each user, both systems will share the same core. The main difference between the two systems is that the developer version will provide more detailed information of the system process whereas the security guard version will have a high level of abstraction in order to keep the usability of the software as high as possible for users without much knowledge of computers.

First we will specify the requirements that both version of the system will share. The following requirements are the core functionality that will be implemented in both versions of the system:

- the system must be able to distinguish the background with the foreground
- the system must be able to find the objects of interest
- the system must be able to classify human and luggage
- the system must be able to track travellers and their luggage
- the system must be able to analyse the behaviours of the travellers and luggage being tracked
- the system must be able to detect abandoned luggage
- the system must be able to give a warning when abandoned luggage is detected.

Depending on the user of the system, additional requirements are needed for the two different version of the system.

The developer version will additionally have the following requirements:

- the system must be able to visualize the various states during the system process.

The security guard version will additionally have the following requirements:

- the system must be able to warn the user using graphics and sound
- the system must be able to write AVI movie files
- the system must be able to give the user the ability to select warnings and provide additional information
- the system must be able to ignore checked warnings
- the system must be able to visualize the end result of the tracking and classifying algorithms.

The *iBlobU Database*

Within the scope of this project we aim to implement a stand-alone database that will allow developers to store information of gathered media files. In this case the *iBlobU Database* system can be used as a catalogue to search for specific data to test the *iBlobU Detection System*. The *iBlobU Database* must have the following functionalities.

1. The developer can insert information of pictures and videos in the database.

2. The developer can change the values of the items in the database.

3. The developer can search the database using the following properties:
   - location
   - the number of luggage in the scene
   - camera brand
   - camera type
   - camera colour settings
   - luggage colour
   - luggage type

- luggage material
- background busyness (crowd or not)
- background dynamic (moving or not).

4. The developer can search the database using key words.
   A keyword interpreter will be implemented. Instead of selecting the properties one by one the user will be able to enter key words in a search bar to search for media files if interest.

5. The developer can view pictures and AVI movies in the database.
   Because the database will consist of images and movie files, the developer user interface will provide functionality to show the images and basic controls to play movie files.

4.4.2 Non-functional Requirements

In the following section we will specify the non-functional requirements which will describe the criteria of the system operation rather than the specific behaviour of the functionality of the system. Unless indicated, the specified requirement accounts for both of the *iBlobU Detection System* and *iBlobU Database*.

4.4.2.1 User Interface and Human Factors

Both the user interfaces of the *iBlobU Detection System* and *iBlobU Database* has to be designed with usability in mind. Because the experience with computers of the end user may vary, it is important that the user interface of both systems can be used by users with little to no computer experience. Assuming that some users may have experience with the Windows Operating System, similar GUI components and styles will be used to meet the expectations of the user. The requirements for the user interface can be summarized as follow:

- the user interface can be used by people who have little knowledge of computers
- the user interface must not be ambiguous and meet the expectations of the user.

Those mentioned requirements are meant for both systems. The *iBlobU Detection System* needs also extra requirements, i.e.:

- the audio or visual warning can be perceived without much effort.
  One of the tasks of the system is to decrease the workload of the user. In the case of the security staff, the system must be able to catch the user attention if he or she is performing other tasks while abandoned luggage is detected.

4.4.2.2 User Manual

In order to provide the user support in operating the system, a user manual will be made. Because personal training and workshop may not be available, it is important that user can learn to use the system individually by reading the system documentation. The provided documentation must be free of jargon and be understandable without any prior knowledge. By reading the documentation of the system the user will understand the possibilities and limitations of the system. The requirements for the documentation are summarized below.

- A user manual has to be written that is understandable for all users

- As user manual has to provide screenshots of the user interface and explain the various components of it
- A user manual has to contain information about the system and how to operate the system.

4.4.2.3 Hardware Considerations

The computer that will run the system has to meet the following requirements:

- Operating System: Windows XP
- Memory: minimum 1 GB RAM
- Processor: minimum 1.8 GHz Intel or an equivalent
- Microsoft .NET 2.0 Framework.

The *iBlobU Detection System* needs not only above requirements, but also:

- When a video files are analyzed, the computer has to have AVI codec in order to view the video file
- The camera which is used to take the frame image must meet the following conditions:

    - colour: 32-bit RGB
    - frame rate: minimum 30 fps
    - resolution: minimum 640 x 480 pixels.

4.4.2.4 Performance Characteristic

The object recognizing process must be performed as rapidly as required by the user or necessitated by the process being controlled, which is in real-time with 5 seconds tolerance delay. The quality of performance depends on the situation of the area in the scene:

The *iBlobU Detection System*

- The best performance will be achieved up to 95% from a scene with less than equal 4 objects
- The best performance will be achieved up to 75% from a scene with more than equal 5 objects (or crowded area).

The *iBlobU Database*

- The given query must be processed within 5 seconds
- The result of the query must be displayed within 5 seconds.

4.4.2.5 Error Handling and Extreme Conditions

In a real life situation the system input are subjected to influence of the environment in which it operates. Although movie files were used during this project, it is important that invalid input to the system are captured and processed to prevent the system from operating correctly. The following requirements are specified to provide a certain level of robustness when the system is dealing with errors and invalid input.

- The system must be robust
- The system must be able to recognize abnormal input

- The system must activate the alarm when video input is not valid.

4.3.3 Pseudo Requirements

Both systems, the *iBlobU Detection System* and *iBlobU Database* has to run on a Windows XP Operating System. The systems are implemented in C# programming language with the Iridium library for supporting matrix operations used in image processing operations.

For the database system the MySQL.net connector will be used to communicate with the database in C#. When video has to be analyzed, it has to be compressed in AVI.

4.3 Environment Analysis

The public transportation environment covers a big surface. The train, the station, the platform and all the places in the public transportation area are included as the environment. A trash bin, a bench, it all counts to objects in the environment. When these objects are not moving by themselves, which they never do, the system can be reasoning detect them as an abandoned luggage. This happens only when the trash bin has been put in the view of the camera suddenly.

5 Software Design

"Luck is the residue of design."
Branch Rickey

Software design is meant to define a solution or planning to the process of building a software system. This can be made after the purposes and specifications are determined by analysing the whole aspects of the upcoming system. First the architecture and model of the software system will be given, followed by the UML of the system.

5.1 Architecture & Model

System architecture is the design or set of relations between the parts of the system. Both the proposed and the *iBlobU* architectures are given, then the *iBlobU* model.

5.1.1 System Architecture

For the Proposed System the architecture overview is as given in figure 14. The architecture shows a data flow of a captured video to the output. The main component in this architecture is the *Abandoned Luggage Detection System*. The image stream reaches this component after it has passed through the *IP Encoder*. The luggage detection and tracking process are all done within this component. Features are retrieved from the *Training Data Server* for the luggage recognition process in the main component. This Server in connected to the database. The output of the main component is visualized to the screen and an audible sound is also given when an abandoned luggage is detected. Also, all captured videos are stored the *Storage Server* for future usage.

As this architecture is for the Proposed System, the architecture of *iBlobU System* is a restricted version, figure 15. Instead a captured camera video, an *AVI Video* is used as input to the system. Also, as stated in Chapter 5, the *iBlobU Database* and the *iBlobU Detection System* are not related. The data flow of the abandoned luggage detection inside the *iBlobU Detection System* is given in more detailed in the next section.

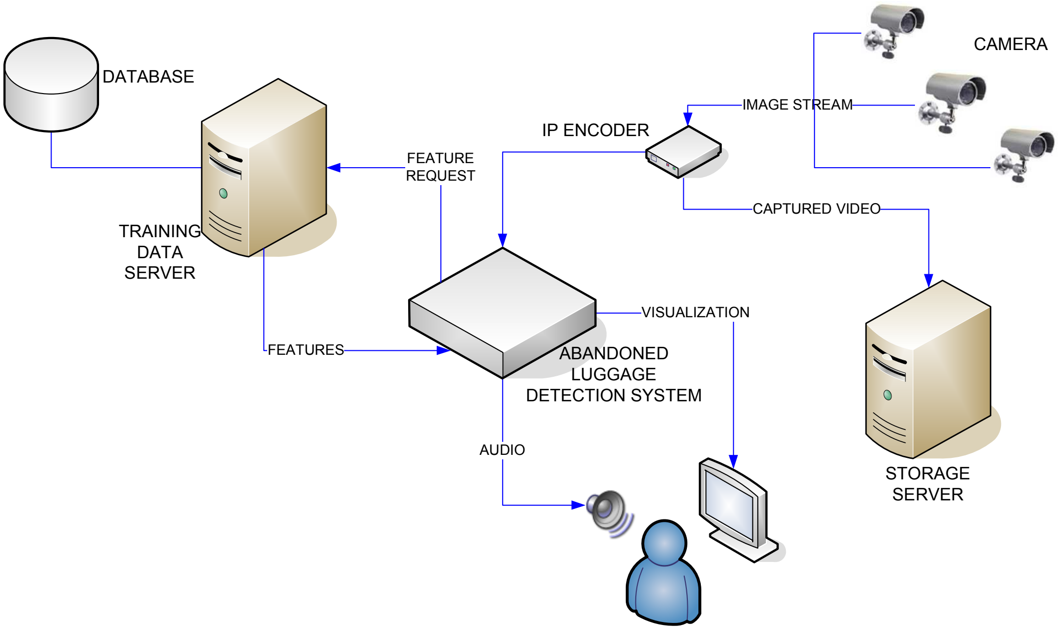


Figure 14: Abandoned Luggage Detection System Architecture

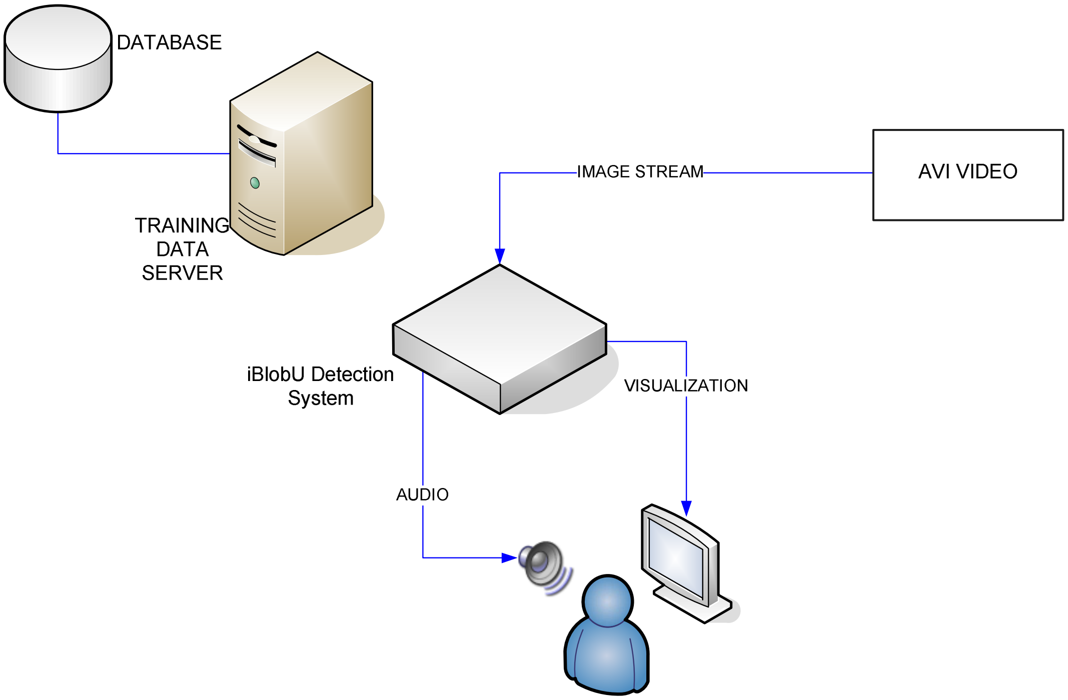


Figure 15: The *iBlobU System* architecture

5.1.2 The iBlobU Detection System Model

Input to the *iBlobU Detection System* for detecting abandoned luggage is a captured image from an AVI Video, as can be seen in figure 16. In order to achieve an ideal image, the background is examined to see whether it is modelled. The various components descriptions inside this system are given below.

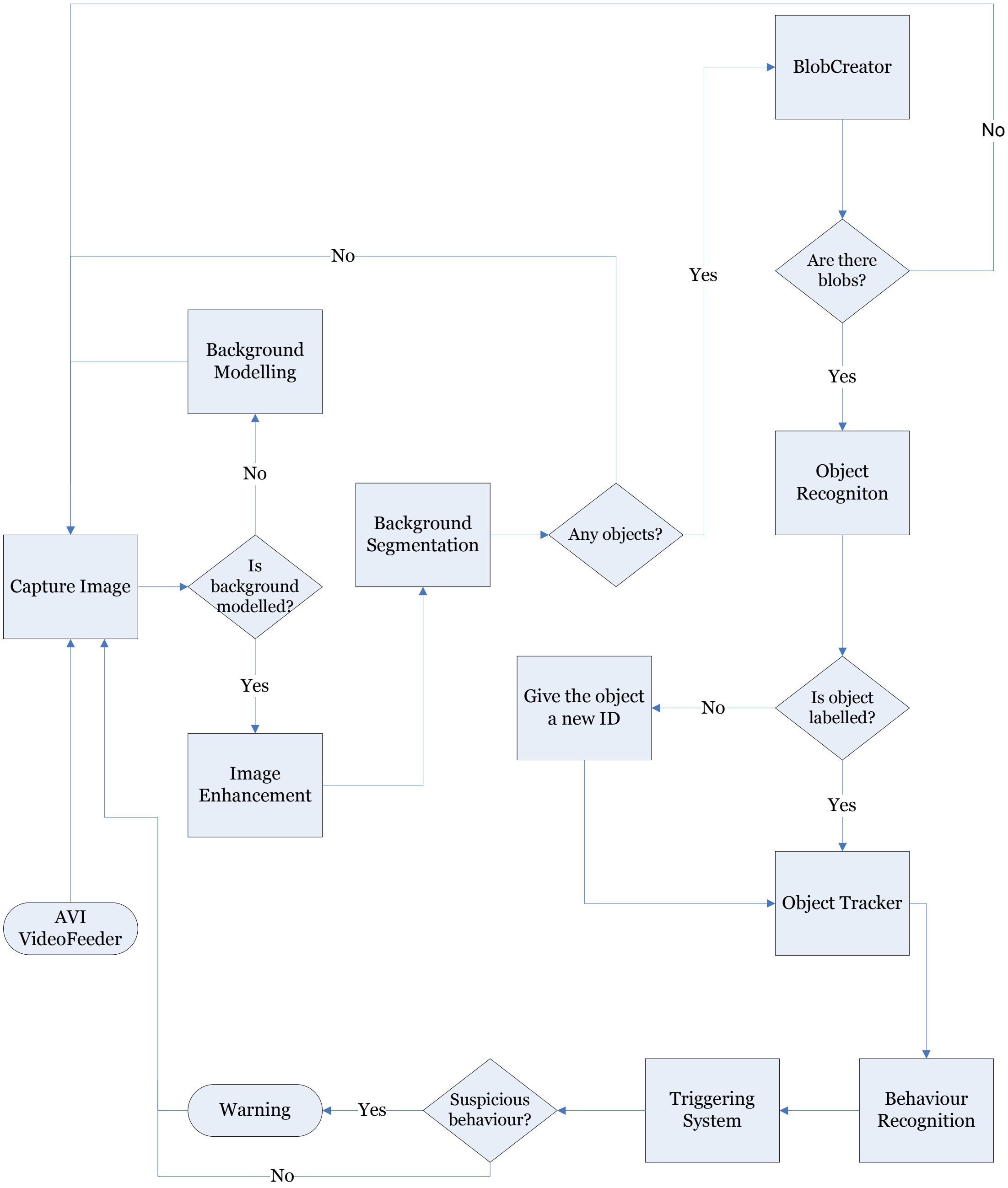


Figure 16: The *iBlobU Detection System* data flow

Background Modelling
This is needed in order to obtain a decent background model for extracting the right objects from the background. This because fast variations cannot be modelled accurately using a single modelled background. Therefore the first hundred frames are used to learn the background.

Image Enhancement
The aim of image enhancement is to improve the quality of the image before further processing. This can be achieved by the following techniques: 2D-Convolution, conversion from RGB channel to greyscale, conversion from RGB channel to normalized RGB, etc. The output image should have less noise and reduce the influence of illumination.

Background Segmentation
In order to extract the objects of interest from the background, the image should be divided into foreground and background. We define foreground as the object(s) of interest which is initially not visible in the scene. By adopting this definition of foreground, the foreground is segmented out of the image given a model of the background. In this way, the background does not have to be considered anymore.

BlobCreator
Having extracted all the object of interest, the objects in this phase of detection are transformed into blobs.

Object Recognition
After the BlobCreator phase, which has just formed blobs without any information about the type of the blobs, it is up to this component (*Object Detector)* to decide which blob is luggage and which one is human.

Object Tracker
This tracker is used to keep track of the recognized objects. The output of this component is given to *Behaviour Recognition* for further processing

Behaviour Recognition
The result of the *Object Tracker* is used to classify and determine the behaviour of the recognized objects according to their movements. There are two behaviour classes, namely, normal and suspicious behaviour. For classifying behaviour, the classifier is trained using training data stored in the *iBlobU Database*.

Triggering System
In this component, a warning is given when an abandoned luggage is recognized and detected.

Warning
Warning is visualized and an audio is also given. Audio warning is divided into levels according to its harmfulness.

5.2 The Unified Modelling Language

UML, Unified Modelling Language, is a system of diagrams that can specify how systems work. It simplifies the software design and communication concerning the design. Therefore, we will discuss the three different diagrams of the system in the next section. First we will discuss the subsystem decomposition diagram, followed by the package diagram. Next the class diagram will be given and the sequence diagrams will be discussed in the last section.

5.2.1 The iBlobU System Decomposition Diagram

The system can be decomposed into three subsystems, the *iBlobU Detection System*, the *Expert System* and the *iBlobU Database*, as shown in figure 17. The subsystems can be implemented separately.

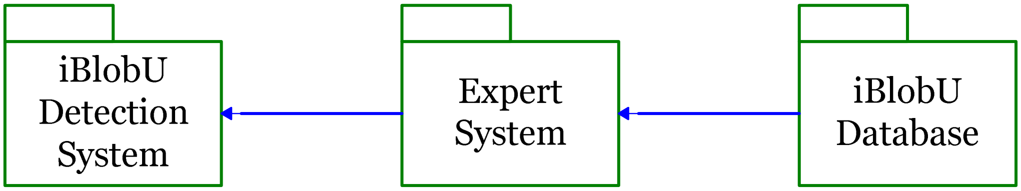


Figure 17: The Subsystem Decomposition Diagram of the system

The *iBlobU Detection System* is the core of the whole system. In this subsystem, the detection, recognition and tracking of an object will be implemented. This subsystem will give a warning when suspicious behaviour or abandoned object is detected. But in order to know how to detect what the objects are, e.g. persons, trash bins, benches and luggage, and what suspicious behaviour is, it is necessary to have some features in order to define various objects and different kinds of suspicious behaviours. For this an expert system is needed. The *Expert System* uses the *iBlobU Database* with the training data. The training data is used by the *Expert System* for acquiring rules. With these rules the *iBlobU Database* can use the features to perform detection on various objects and suspicious behaviour and it can give a warning when necessary.

5.2.2 The iBlobU System Package Diagram

The package diagram goes one level deeper into the design than the subsystem decomposition diagram does. In the diagram, the packages of each subsystem and the connection between them are shown. This can be seen in figure 18 and figure 19.

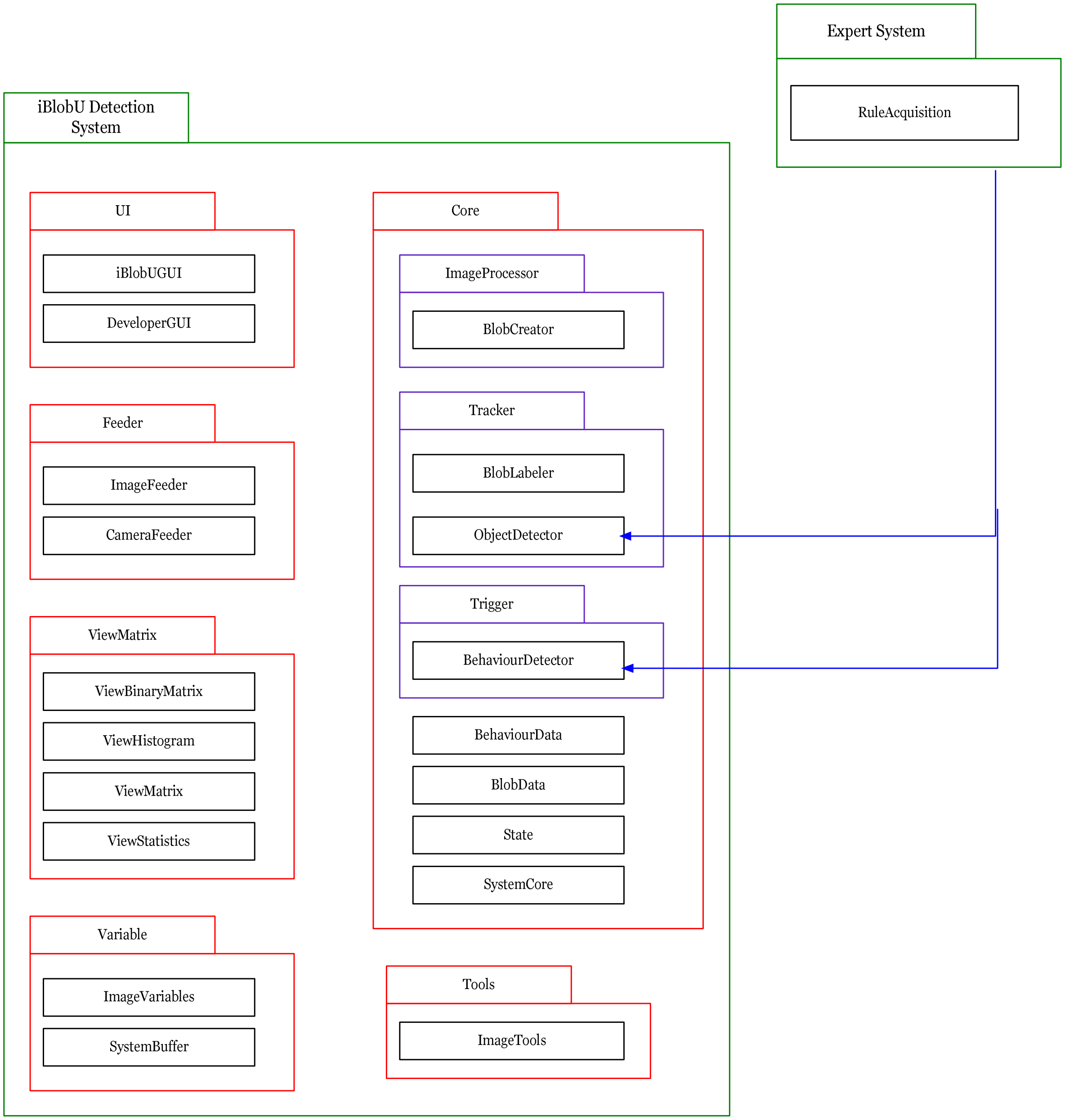


Figure 18: The Package Diagram of the *iBlobU Detection System* connected to Expert System

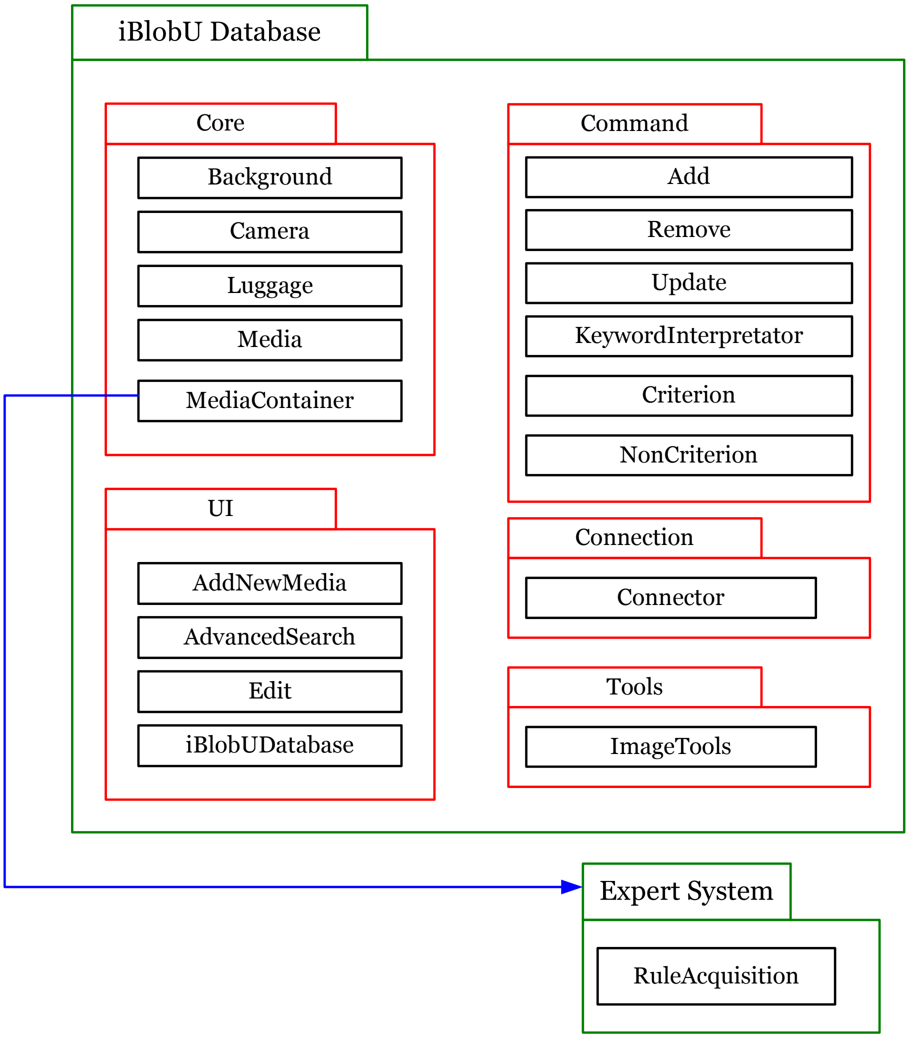


Figure 19: The Package Diagram of the *iBlobU Database* connected to Expert System

Some (sub)-packages belong to another package, e.g. *Tracker* and *Trigger* in *Core* package from the *iBlobU Detection System*. The *iBlobU Database* has a connection with Expert System through the *MediaContainer* in *Core* package. The *Expert System* contains the features to define the objects and suspicious behaviours, as mentioned above. These features are further needed by some classes in *Tracker* and *Trigger* package and thus these two sub-packages are connected with the *Expert System* to be able to access the features.

The *Variable* Package has a function as a storage or buffer for the constant values and (list of) variables. Its classes are constructed as static classes to give every class access to those values in *iBlobU Detection System*. *Tools* and *ViewMatrix* packages are also made to facilitate other packages. *ViewMatrix* package is a library with various matrix functions, i.e. matrix properties (max, min, sum, mean, variance, etc), matrix to/from bitmap conversion, matrix convolutions, morphological operation and other matrix operations that are useful to assist the other packages.

As in the *iBlobU Detection System*, the *Core* package in *iBlobU Database* contains the main classes and *Tools* package contains image tools to set the images into the right size to be displayed. But differently for the *UI* package from the Database, which actually contains only one main interface with sub-window interfaces while for the system contains two interfaces for different purposes, for the developer and end-user.

The communication between *iBlobU Database* and the *MySql Database* server using *MySql Connector* is obtained by *Connection* package. This package is set to static to allow all commands in the *Command* package to make the connection with *MySql Database*.

5.2.3 The iBlobU System Class Diagram

The class diagram is the mainstay of object-oriented analysis and design. It shows the structure of the classes in the *iBlobU System*. The interrelationship between the classes will be discussed in this section. The proposed system class diagram in depicted in figure 20 and figure 21. The first figure shows the class diagram of the *iBlobU Detection System* in connection with the *iBlobU Database*. The connection between *ObjectDetector-RuleAcquisition* and *BehaviourDetector-RuleAcquisition* will not be implemented as the *iBlobU Detection System* and *iBlobU Database* is implemented individually, as stated earlier. Figure 21 shows the rest of the class diagram of the *iBlobU Database*.

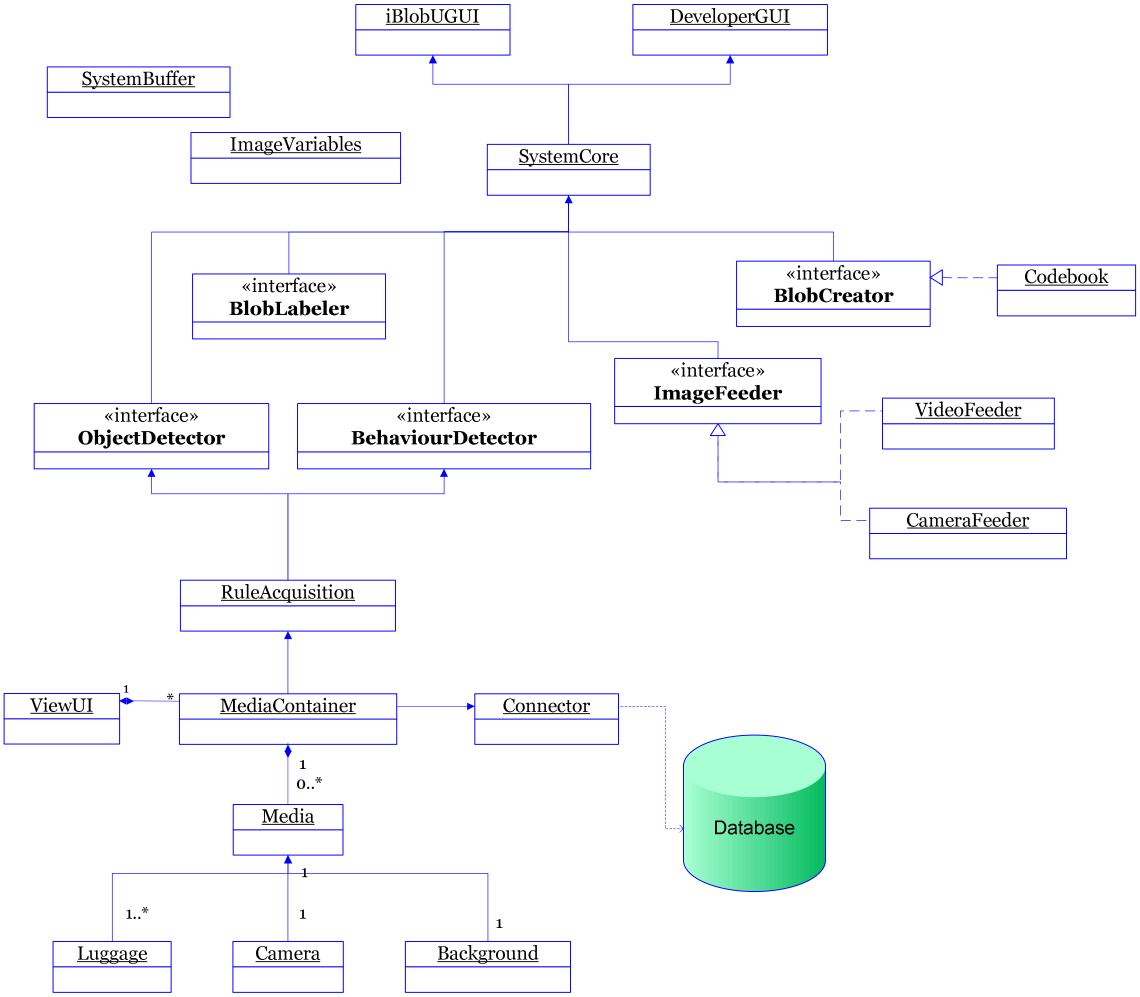


Figure 20: The Class Diagram of the *iBlobU System*

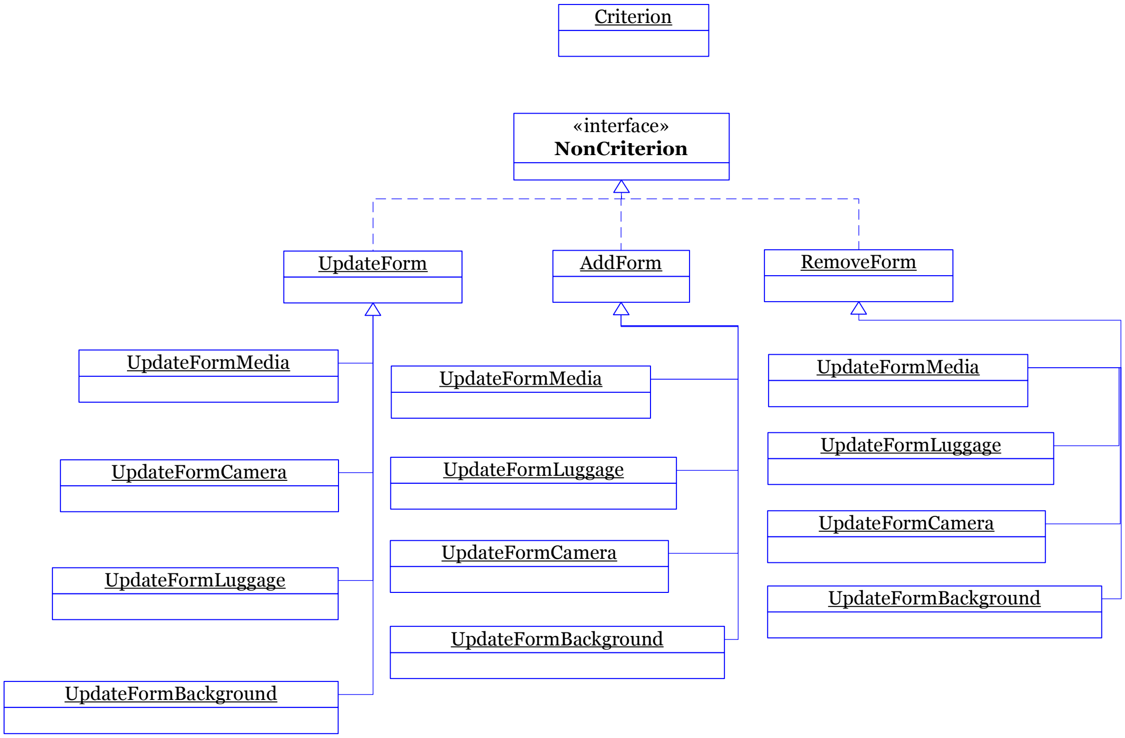


Figure 21: The Class Diagram of the *iBlobU Database*

The *iBlobU System* can be represented by two different Graphical User Interfaces, namely *iBlobU* GUI for the end-user version and Developer GUI for the developer or researcher version. The GUIs are divided in two versions because there is some functionality for the developer which are not needed or not the interest of the end-user. These GUI classes are generated by the main class of the *iBlobU Detection System*, *SystemCore* class. The *SystemCore* class has a connection to the *ImageFeeder, BlobCreator, BlobLabeler, ObjectDetector* and *BehaviourDetector* interfaces and contains the references from the whole process in the *iBlobU Detection System*.

Some of classes in the class diagram are designated with inheritance usage in considering of techniques possibilities to the solution for the system problem or because the children of parents class share the same properties or methods. So, this decision is made for the convenience of programming. A few techniques can inherit the *BlobCreator* interface, e.g. *CodewordConstruction* which creates a model for the background by assigning the background pixel according to that model. *ImageFeeder* interface inherits an image capture to *CameraFeeder* and *VideoFeeder* classes to grab each frame from the camera/video streaming and to open/close the streaming connection.

The *iBlobU Detection System* is related to the *iBlobU Database* since the *Object Detector* and *Behaviour Detector* use *Rule Aqusition* to get the data automatically from the Database and use it as training data to distinguish the object of interest as people or luggage and to recognize suspicious or normal behaviour.

The main classes of this *iBlobU Database* are *Luggage, Camera* and *Background* which contain the properties of corresponding object and are carried by the *MediaContainer*. The *Connector* static class is responsible for the connection with the *MySQL Database* and allows *Criterion* and *NonCriterion* to use the connection in order to send the queries to the database.

5.2.4 The iBlobU System Sequence Diagram

The sequence diagram represents the communication between objects. It visualizes the functions and messages which are exchanged between objects. First the sequence diagram of the *iBlobU Detection System* will be described followed by the sequence diagrams of the *iBlobU Database*. Only the trivial sequence diagrams are given and described. The sequence diagram in figure 22 shows how the classes of the *iBlobU Detection System* behave in detecting an abandoned luggage. Each operation step is followed by another; therefore, each class need to be carried out carefully.

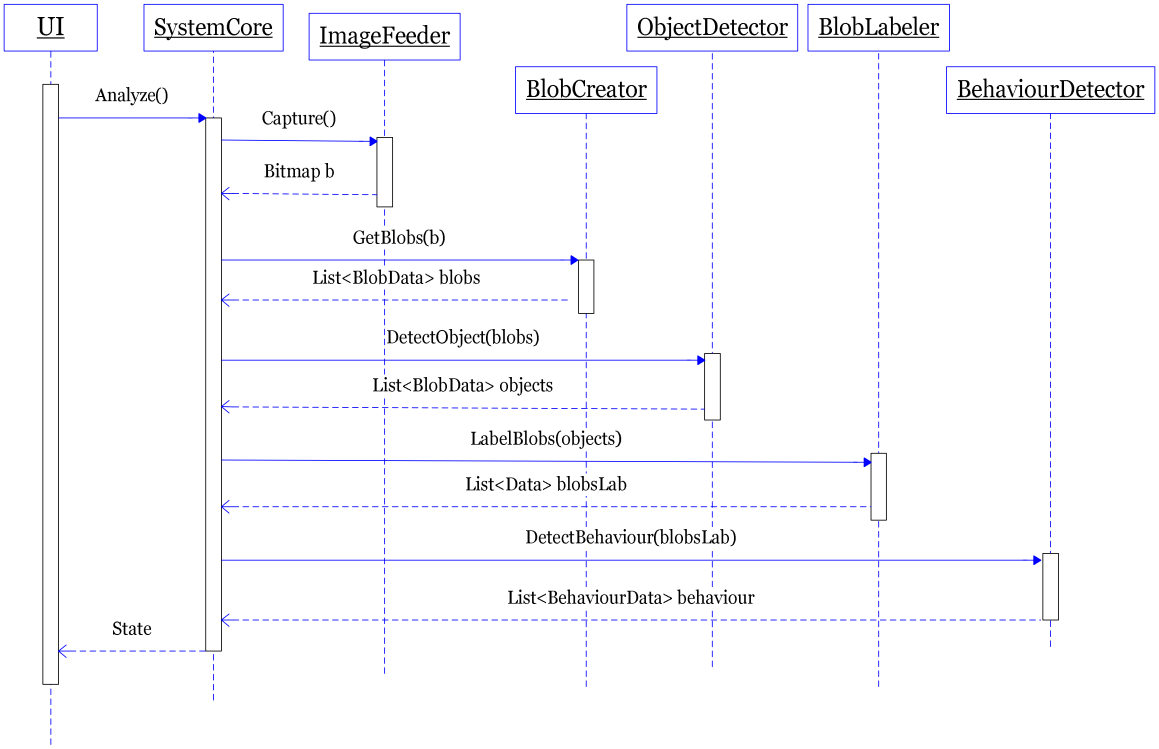


Figure 22: The Sequence Diagram of the *iBlobU Detection System*

To detect an abandoned luggage, the system has to analyze all objects in each frame of the video. This analysis is handled by the *SystemCore*. By sending a *Capture* message to the *ImageFeeder*, a frame of the video will be captured and transformed into a *bitmap image*. The *bitmap image* will be returned to the *SystemCore* for further analyzing. Having the *bitmap image*, all the objects of interest will be extracted from the background. Also, a list of Blobs will be created by using these extracted objects. This Blobs developing procedure is done by *BlobCreator* in which the *SystemCore* has to send a *getBlobs* message first.

After constructing all the existing Blobs, the responsibilities of the detecting process are in the hands of the *Object Detector* now. Its task is to determine which blob is human and which is a luggage. The *SystemCore* will be receiving a list of Blobdata in which each blob is constructed as object containing its type. With this, the next step in de sequence is to track each detected object. This is done by sending a *LabelBlobs* message to the *BlobLabeler*. Therefore, the *BlobLabeler* give each detected object a unique ID number. It also checks whether ID-number of the detected object is assigned correctly when certain objects are merged or split.

The final step in this sequence is detecting the behaviour of each labelled object, in charge by the *Behaviour Detector*. As a result, each object's behaviour is listed and sent to the *SystemCore* in which the state of the frame will be displayed in the UI. If the state is abandoned, a warning/alarm will be given, otherwise nothing will happen.

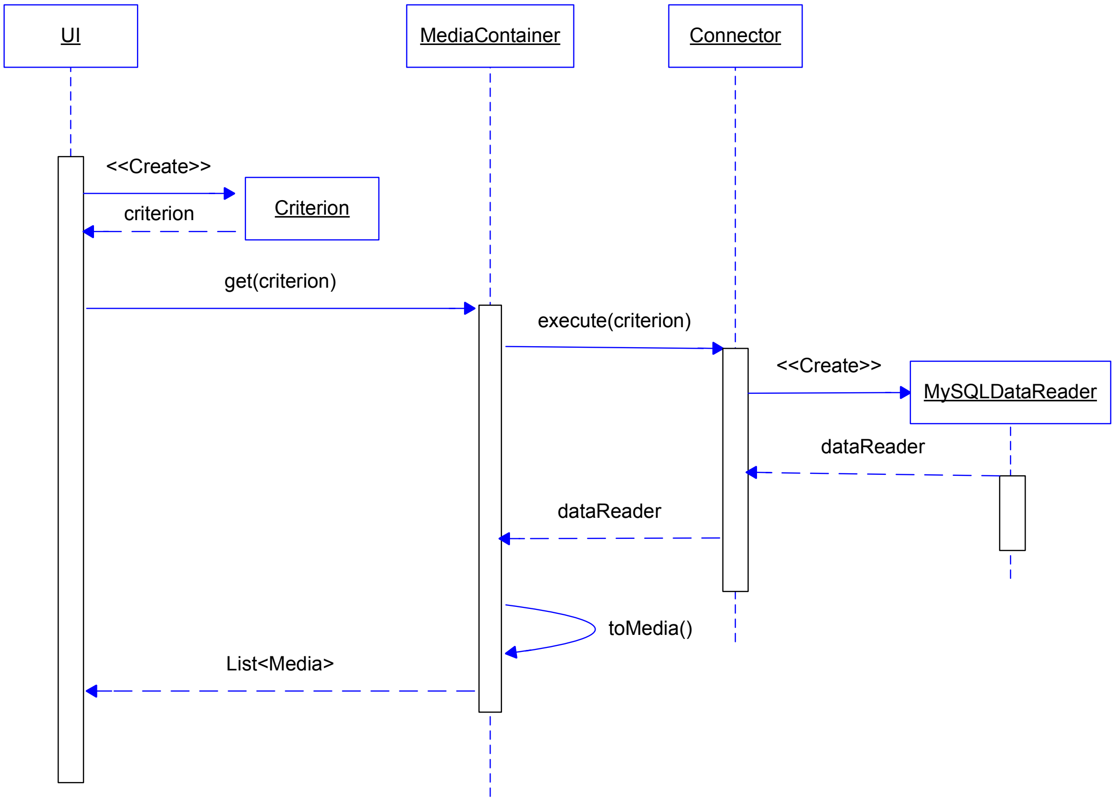


Figure 23: Sequence Diagram of media retrieval

In figure 23, we can see the flow of events that happen when the user wants to retrieve media from the Database. First the *Criterion* mode is activated by the user. A list of predefined criterions will then be returned to the user, where he or she can create their own queries by filling the criterions in the criterion list. After this has been done and the search button has been pressed, a message *get (criterion)* is sent to the *Media Container*. This container maintains the *Connector* which is the only object that communicates with *MySQLDataReader*. Once the Connector has received the requested data from the *MySQLDataReader*, it will be converted to a media object by the *MediaContainer*. After this conversion, the requested media will be shown on the user interface, observes by the user.

To add, remove or update media in the Database (figure 24), the sequence procedure is pretty much the same. First the user will receive a form where information can be filled in which the user wishes to update, add or remove. This form is fetched by the *MediaContainer* by sending the messages C*reate* to *NonCriterion*. After all necessary information is inserted by the user; the information will be executed by *Connector*. First the Connector will send a *toCommand* message to *NonCriterion* in order to fetch the information transformed into a query. Finally this will be executed and the result will be returned to the user to indicate whether the action has succeeded.

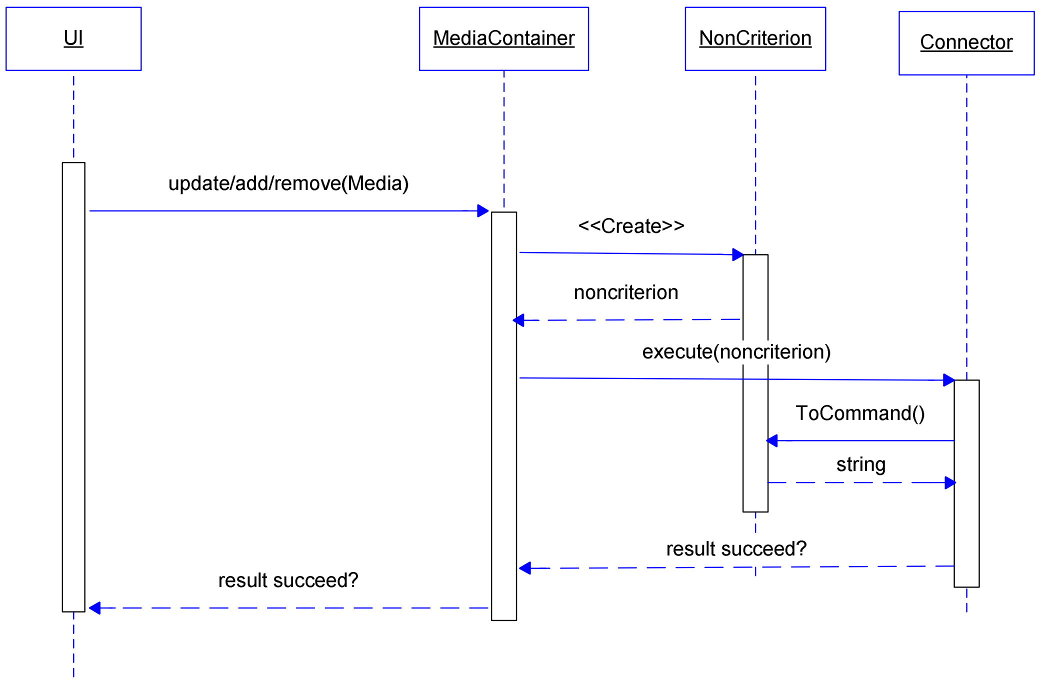


Figure 24: Sequence Diagram for media update/ addition/ removal

6

System Implementation

> *"At the present rate of progress, it is almost impossible to imagine*
> *any technical feat that cannot be achieved - if it can be achieved at all –*
> *within the next few hundred years."* Arthur C. Clarke

This chapter focuses on the details of implementation which are applied in our software in the scope of this "Abandoned Luggage Detection" project. As stated in the previous sections, there will be two independent systems implemented in this project. They are:

- *the iBlobU Database*
- *the iBlobU Abandoned Luggage Detection System.*

The whole project is implemented in Microsoft Visual C# language which is supported by .NET Framework 2.0. The choice that was taken in consideration to implement in this programming language is:

- Visual C# is an 'easy to learn' object-oriented programming language
- Visual C# provides a huge development environment and is the most decent programming language to implement an application in Windows Platform
- Visual C# is well-supported and also well-documented
- Visual C# makes it possible to code in a managed environment which will avoid 'memory leak' in the software
- Visual C# is the programming language which all group members of this project are experienced in.

To assist in programming the software's, Microsoft Visual C# 2005 Express Editions is utilized as the Integrated Development Environment (IDE).

6.1 The *iBlobU Database*

The *iBlobU Database* is an application where the user can add, edit, view and remove the information of the image or video. The information of media which is kept by the database is the information which is manually filled by the user beforehand. This information is similar to MPEG-7 multimedia content description standard where the description is associated with the content itself.

The information of a media which the user has to fill in is:

1. Media properties:

    - type of the media : audio or video
    - width of the image / frame of the video
    - height of the image / frame of the video
    - size of the media file
    - location where the image or video is recorded
    - time when the image or video is recorded (Morning, Noon, Evening)
    - the compression of media file
    - number of luggage which are in the visible in the media
    - filename of the media
    - file path of the media
    - description of the content of media.

2. Camera properties:

    - brand of the camera
    - type of the camera
    - focal Length of the camera
    - colour Setting of the camera (Greyscale or Colour).

3. Luggage properties:

    - colour of the luggage
    - type of the luggage
    - material of the luggage
    - shape / Geometry of the luggage
    - size of the luggage (Small, Medium or Large).

4. Background properties:

    - the busyness in the media (Quiet, Normal, Busy)
    - dynamic in the media (Static or Dynamic).

In this system, the database will be saved in online MySQL server. The MySQL is chosen as the Database Management System (DBMS) because:

- there are lots of libraries available for accessing MySQL database in Visual C# which makes the coding of database communication less complicated
- the MySQL is the DBMS which the group members of this project are experienced in.

6.1.1 Tables Definition

To save the information of the image or video in a structured way, four different tables are created. These are:

1. media table
2. camera table
3. luggage table
4. background table.

Each of the table will have a unique id which is the primary key of each row in the table. By joining these four tables, the complete information of a media can be acquired. The structure of each table is as follows:

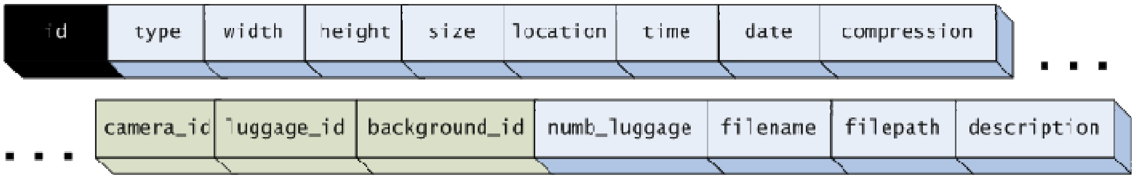


Figure 25: Database media table

Table 5: Data dictionary of media table of *iBlobU Database*

Field	Type
id	int(11)
type	enum('Photo', 'Video')
width	int(11)
height	int(11)
size	double
location	varchar(50)
time	enum('Morning', 'Noon', 'Evening')
date	date
compression	varchar(50)
camera_id	int(11)
luggage_id	int(11)
background_id	int(11)
numb_luggage	int(11)
filename	varchar(50)
filepath	varchar(300)
description	text

As the name suggests, this table contains the properties of the media. Field 'id' is auto-incremented and always unique. This field is the primary-key of this table. This table has also 3 foreign keys which are responsible in connecting with the camera table, luggage table and background table. These three foreign keys are:

1. camera_id
2. luggage_id
3. background_id.

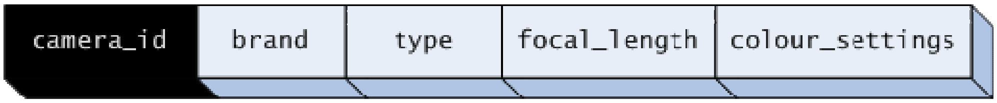


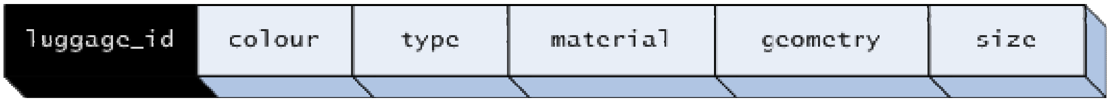
Figure 26: Database camera table

Table 6: Data dictionary of camera table of *iBlobU Database*

Field	Type
camera_id	int(11)

brand	varchar(50)
type	varchar(50)
focal_length	double
colour_settings	enum('Colour', 'Grayscale')

In this table, field 'camera_id' is the primary-key of the table. This table is created to avoid redundancy in saving camera properties.



luggage_id	colour	type	material	geometry	size

Figure 27: Database luggage table

Table 7: Luggage table of *iBlobU Database*

Field	Type
luggage_id	int(11)
colour	varchar(50)
type	varchar(200)
material	varchar(200)
geometry	varchar(200)
size	enum('Small', 'Medium', 'Large')

In this table, field 'luggage_id' is the primary-key of the table. This table is created to avoid redundancy in saving luggage properties.

background_id	busyness	dynamic

Figure 28: Database background table

Table 8: Background table of *iBlobU Database*

Field	Type
background_id	int(11)
busyness	enum('Quiet', 'Normal', 'Busy')
dynamic	tinyint(1)

In this table, field 'background_id' is the primary-key of the table. This table is created to avoid redundancy in saving background properties.

6.1.2 MySQL Connector/NET

To communicate with MySQL server in Visual C#, MySQL Connector/NET is used. MySQL Connector/NET is library which enables the developers to create .NET application that require connectivity with MySQL. This connector does not need any ‘MySQL client library’ to be used which makes it possible to run our database system without any installation of other third-party software.

To open a connection with the database, a MySQLConnection object has to be created. It can be done with these source code lines:

```
private static readonly String connectionString =
      String.Format("Database = {0}; Data Source = {1}; User Id = {2};
                     Password = {3}", "bsc_luggage",
                    "cpsql.twi.tudelft.nl", "bscluggage", "**********");

private static MySqlConnection dbConnection = new
      MySqlConnection(Connector.connectionString);
```

After the connection is made, it can be used to execute any query or non-query command. To create a command, the source code will be:

```
MySqlCommand myCommand = new MySqlCommand(queryString, dbConnection);
```

where queryString is the MySQL command line and dbConnection is the MySQLConnection object.

Before MySQLCommand can be used to execute commands, it must open the connection by calling this method:

```
myCommand.Connection.Open();
```

Then, to execute a SEARCH, UPDATE, INSERT or REMOVE command this method has to be invoked:

```
MySqlDataReader myReader = myCommand.ExecuteReader();
```

In case it is a SEARCH command, this method will return a MySqlDataReader object which contains all the field's values of the search result.

After each command, the connection is closed to reduce any unnecessary network connection. This can be done with this piece of source code:

```
myCommand.Connection.Close();
```

6.2 The *iBlobU Abandoned Luggage Detection System*

The *iBlobU Abandoned Luggage Detection System* is a application which is able to analyze AVI video or sequence of images grabbed by webcam and give a warning when it's established that a baggage is abandoned. In the scope of this project, the performance is optimized to be able to detect abandoned in the bench of the train station.

In this section, techniques and algorithms which are applied during the implementation of the application will be discussed.

6.2.1 ViewMatrix Library

Implementing computer vision application can be really complex without an easy learnable image processing / computer vision library. In the beginning of the project, the choice of the library has to be made and one criterion which this library should satisfy is that it must reduce the workload during coding the software. After searching some existing image processing library which work with Visual C# programming language, these libraries come into considerations:

1. OpenCV dotnet Wrapper (http://code.google.com/p/opencvdotnet/)
   This library contains of Dynamic Link Library (DLL) written in Managed C++ that wraps OpenCV library in .NET classes. In this manner, this library can be applied in our Visual C# coded software. OpenCV is at this moment the most popular computer vision library which guarantees the quality and performance of this library.
   Drawbacks of this library are:
   a. It does not provide all functions of OpenCV. To add new function to the wrapper, source codes have to be written in Managed C++ to wrap the OpenCV function which will demand more workload.
   b. Available documentation and examples are too few
   c. To make it work properly, the user has to run installer of this library and it does not always work in all computers. This will make the application less flexible.

2. AForge .NET (http://code.google.com/p/aforge/)
   AForge.NET is a C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence (image processing, neural networks, genetic algorithms, machine learning). To use this library, it is not necessary to run the installer, which will make the application more flexible. On the web, there are also many examples in using this library and the documentation of this library is also available to download.
   Drawbacks of this library are:
   a. All operations are done in Bitmap object (an object in .NET framework which contain a bitmap image), which makes it more difficult to do a pixel-based operation.
   b. The performance of this library is slow.

After considering the drawback and benefit and testing these libraries, still these libraries have minimal impact in reducing the workload of implementation.

Since most of group members are familiar with MATLAB, an ideal library will be a library which represents the image as matrix and also provides the corresponding matrix operations such as multiplication, subtraction, etc. By having the image as matrix or an array of matrix, it will also be easier to do pixel-per-pixel operation. This ideal library should also be able to run image processing functions such as 2-D convolution between two matrixes.

In order to reduce the workload of coding and introduce a more familiar way of coding through the project's implementation, we have decided to make our own library which is able to do matrix operations and simple image processing function. The name of this library is ViewMatrix and three most important classes are:

- **Matrix**
  An instance of this class contains a 2-D array of the floating point. This is where the value of every number in a matrix is kept. This class is also equipped with matrix operations such as adding, subtracting, matrix multiplying, array multiplying, matrix dividing, etc. Furthermore, this Matrix gives access to get or set one number in a matrix, which is very favourable when a pixel-per-pixel operation has to be done.

An example of the usage of this class:

```
Matrix a = new Matrix(100, 100, 1);
Matrix b = a + a;
b = b * a;
b = Matrix.ArrayMultiply(a, b);
```

As seen in the code example above, the code style is similar to how coding in MATLAB is done.

- **ViewMatrix**
  ViewMatrix is class which has a collection of static methods to process Matrix object. These methods are mostly image processing operations and conversion between Bitmap object to Matrix object and vice versa. Some image processing operations which are implemented in this class are 2D-Convolution, conversion from RGB channel to greyscale, conversion from RGB channel to normalized RGB.
  An example of the usage of this class:

```
Matrix[] rgb = ViewMatrix.ViewMatrix.Bitmap2Matrix(bitmap);
Matrix gray = ViewMatrix.ViewMatrix.RGB2Grayscale(rgb);
Bitmap bmp =
      ViewMatrix.ViewMatrix.Matrix2GrayscaleBitmap(gray);
```

- **ViewBinaryMatrix**
  Thresholding an image is an operation which is often to be performed when processing a matrix. When a matrix is thresholded, the value of matrix has to be converted into Boolean value and this class provides the matrix with Boolean value. An instance of this class contains a 2-D array of Boolean value. This class has also logical operations such as NOT, OR, AND, GREATER THAN, and LESS THAN. There are also morphological operations such as dilation and erosion.
  An example of the usage of this class:

```
ViewBinaryMatrix threshold =
      ViewBinaryMatrix.GreaterThan(matrix,
      ImageVariables.Threshold.THRESHOLD_VALUE);

threshold = threshold.Dilation(ViewBinaryMatrix.BinaryKernel(3,
      ViewBinaryMatrix.nNeighbourhood.EightNeighbourhood));
threshold = threshold.Erosion(ViewBinaryMatrix.BinaryKernel(5,
      ViewBinaryMatrix.nNeighbourhood.EightNeighbourhood));
```

This library is applied in the *iBlobU Abandoned Luggage Detection System* in order to do image processing as matrix operations. Its similarity with matrix operations in MATLAB has made us easier to code and experiment during the project.

6.2.2 Background Subtraction

The first step in detecting abandoned luggage is separating between foreground and background. In this case, we define foreground as the object(s) of interest which is initially not visible in the scene and the rest of the image which is not a foreground is defined as background. In the train or train station, the foreground is usually passengers and their baggage. The background is all non-moving things such as wall, bench, floor or trash bin and if the place is located outside a building, the background also includes sky and trees with moving leaves.

By adopting this definition of foreground, the foreground can be segmented out of the image given a model of the background. To perform such segmentation, background subtraction techniques are chosen to be applied. The assumption, that a model of the background has to be acquired beforehand, becomes an essential constraint in the implementation of the *iBlobU Abandoned Luggage Detection System* with the consequence that the system has to be fed with initial images without any foreground objects in the scene to model the background. In the course of this project, two techniques of background subtraction are implemented.

6.2.2.1 Naïve Approach

This technique calculates the difference of brightness between two pixels in the same location for all pixels in the image. The details of this technique are discussed in the chapter 3.

In the beginning of the application, it is assumed that the first frame will be the image without any foreground objects. This frame is taken as the model of the background and used in segmenting foreground for the rest of the frames. The threshold value to distinguish between foreground and background is determined manually after doing lots of testing on various movies. The result of the segmentation using this technique is depicted in figure 29.

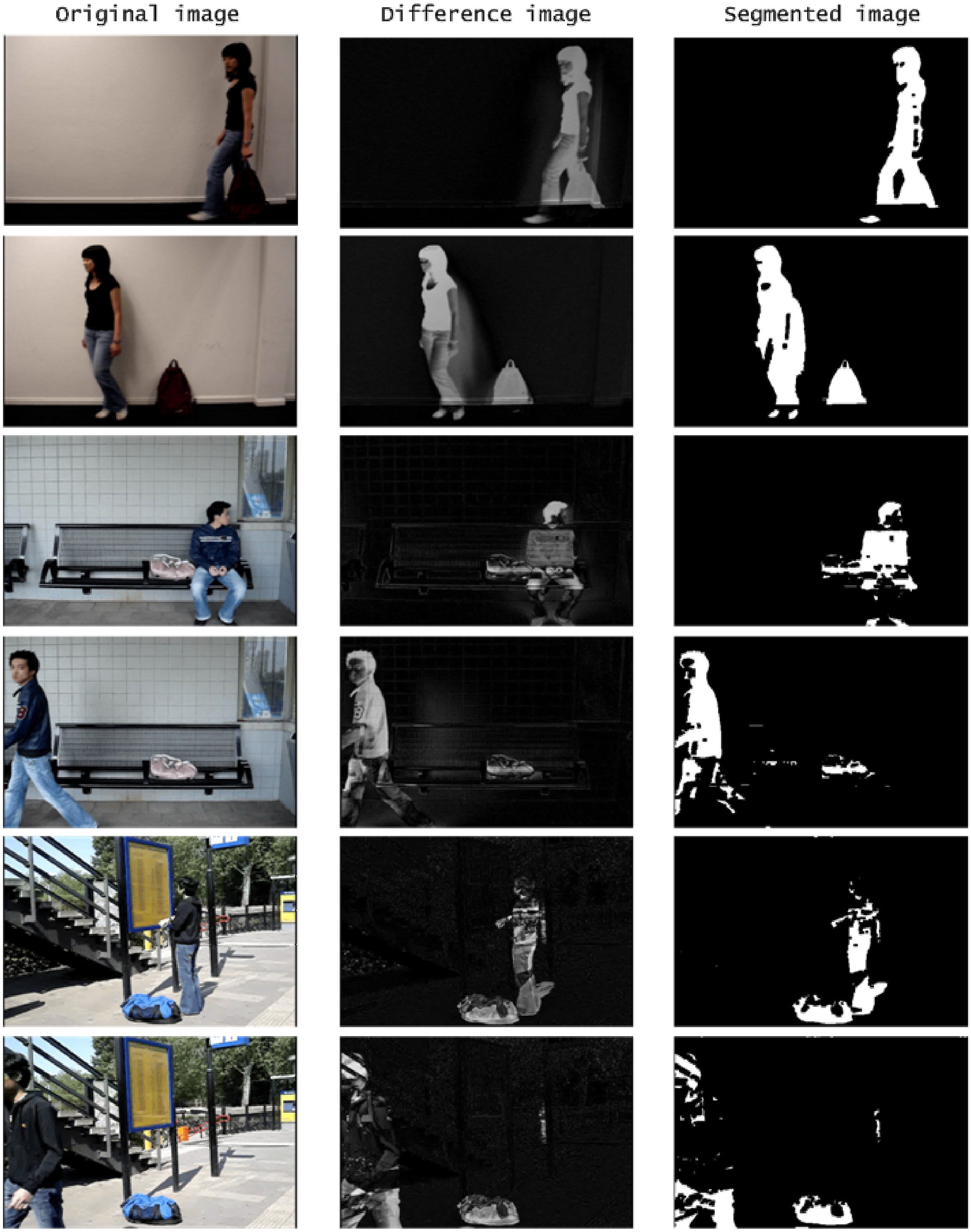


Figure 29: The result of the background subtraction using Naïve Approach

Since the result of the segmentation is a binary image, morphological operations such as dilation and erosion can be run to refine the edge of the segmented object.

This technique works well in a scene where the background is not complex and the contrast between foreground and background is high. But in the environment which usually has a complex background like in the train station, it does not segment the foreground robustly. It is usually caused by noise and luminance changes. Another drawback of this system is that it uses only brightness to compare the pixel which makes it impossible to distinguish the foreground with the background when the background has a dark colour like blue and the foreground has also a dark colour like dark red. An example of the segmentation done with this technique with a complex background can be seen in the row 3, 4, 5 and 6 of figure 29. As seen in the figure, the faces of the passengers are not detected as foreground which is not correct. We also see that the segmented upper body of the passenger is not connected anymore with the lower part of the body. To fix this, the colour information has to be used to segment the foreground and modelling the background must be made more advanced. The next technique is an attempt to segment foreground of an image with a complex background.

6.2.2.2 Background Subtraction by Codebook Construction

The details about Background Subtraction by Codebook Construction can be read in the section 3.1.2 of this thesis. To use this technique, it begins with modelling of the background by means of the initial frames where there are no foreground objects visible. In the implementation of *iBlobU Abandoned Luggage Detection System*, the first 50 frames will be used as the initial frames to model the background.

Compared to brightness difference technique, this technique also takes colour information into consideration to determine whether a pixel is a foreground pixel. The calculation of the difference for colour and brightness can be found in the equation Eq.3.6 and Eq.3.9, respectively. The result of Background Subtraction by Codebook Construction can be seen in figure 30.

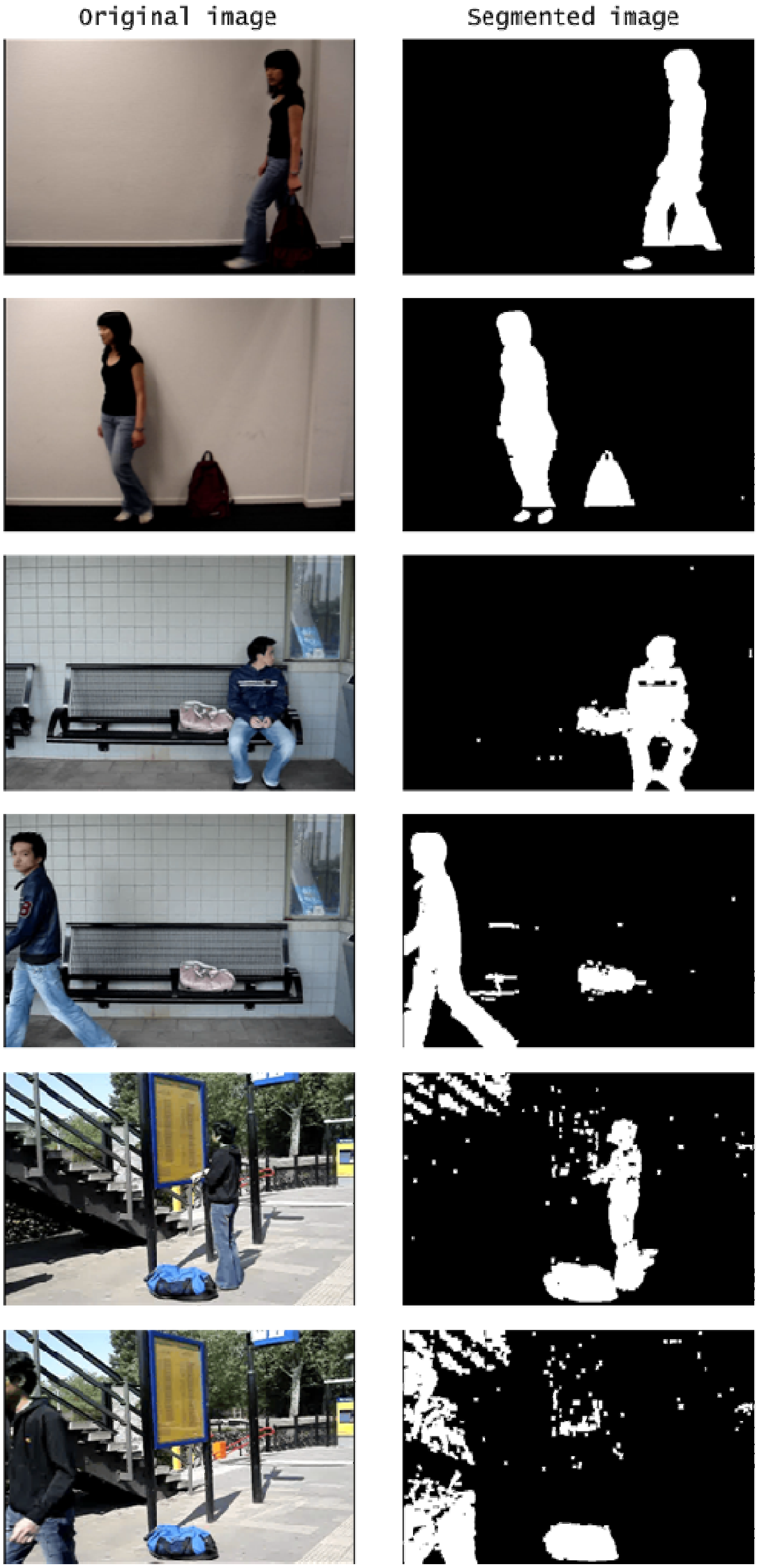


Figure 30: The result of the background subtraction using Codebook Construction

As we see in the figure above, this technique can handle complex background. In contrary with the first technique, the faces of the passengers are also detected as foreground. The bodies of the passengers are made nicely as one blob which is favourable for further processing. In the segmented images, noises are sometimes present but this can be removed by ignoring the blobs which have only few amounts of pixels.

After comparing both of the results, the Background Subtraction by Codebook Construction is finally chosen to be applied in the *iBlobU Abandoned Luggage Detection System*. The main consideration in choosing this technique is because it uses colour information to segment the image. Also, an object in the image usually result as one or two big blobs in the segmented image, this is really favourable for the next processing steps in the iBlobU Abandoned Luggage Detection System.

6.2.3 Human-Bags Classifier

After the binary image is obtained by background subtraction, the blobs in this image are retrieved. In these blobs, there are usually blobs which contain few amounts of pixels which in the most cases represent noise or ignorable objects. If these kinds of blobs are found, it will be ignored and will also be left unprocessed. The rest of blobs will go through the human-bags classifier in order to give these blobs a semantic meaning.

To classify the blob as an object there are three features that are used:

1. The size of the blob
   In this abandoned luggage detection system, it is assumed that a human has usually bigger size than a bag. The size of the blob is calculated by counting the amounts of pixels in a blob. See figure 31 for the illustration of this assumption.

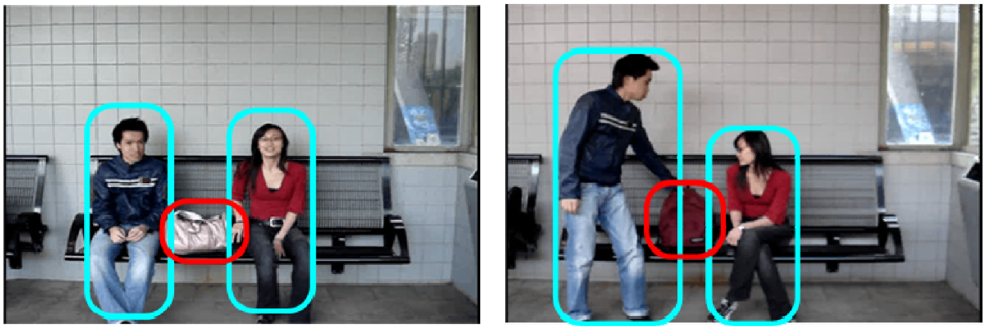


Figure 31: Bags and human in the image

   In this figure, it can be seen that the blue round rectangle that represents human is always bigger than the red round rectangle that represents bag.

2. The amounts of skin pixels in the blobs
   A human can be classified by checking whether an object has skin colour pixels. Faces and hands of humans are usually uncovered with clothes, which will make it easier to distinguish human from bags which rarely have skin colour.

   To detect skin colour, the Human Skin Colour Clustering method, which is explained in [11], is applied. This method defines explicitly the boundaries skin cluster in RGB colour space, which is (R, G, B) is classified as skin if:

$$R > 95 \textit{ and } G > 40 \textit{ and } B > 20 \textit{ and}$$
$$\max(R, G, B) - \min(R, G, B) > 15 \textit{ and}$$
$$|R - G| > 15 \textit{ and } R > G \textit{ and } R > B$$

The result of this method is depicted in figure 32.

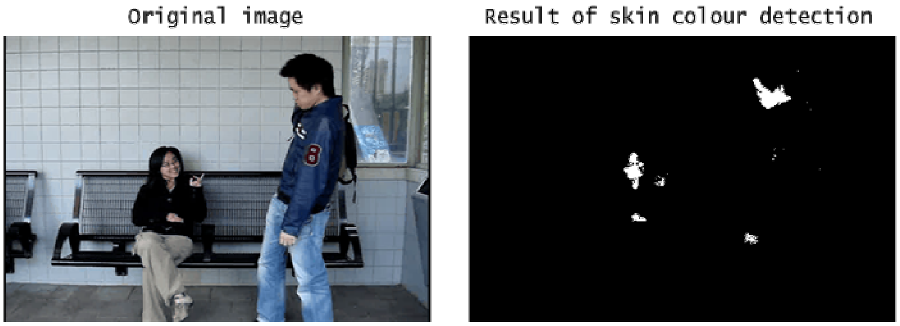


Figure 32: The result of skin colour detection using Human Skin Colour Clustering

3. The silhouette of the blob
The details of object classification based on silhouettes are explained in chapter 3.2. The templates which are used to classify the objects are made manually beforehand. In this Abandoned Luggage Detection system, there are 3 templates used. They are:

    - human group template
    - human template
    - bags template.

    Each template contains the distance signal features of the silhouettes which belong to the object that this template represents. The examples of silhouettes which are used in the templates can be seen in the figure 33.

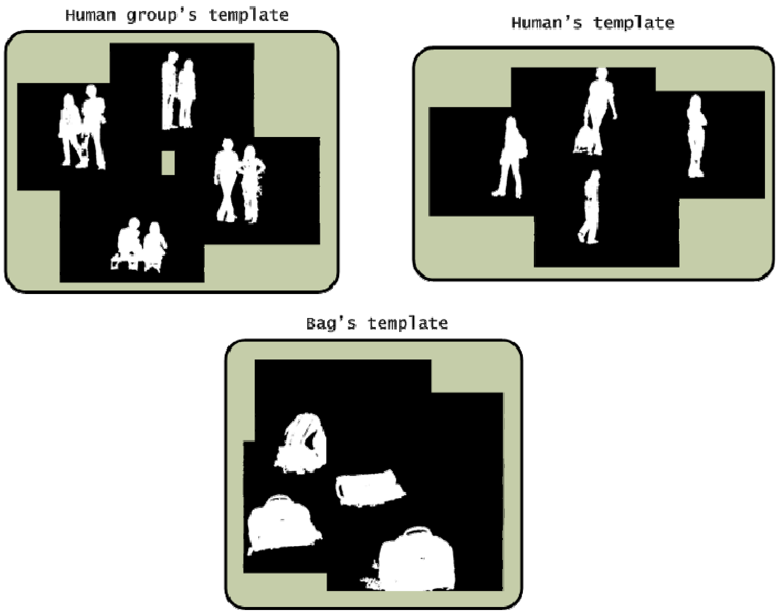


Figure 33: Templates which are used in *iBlobU Abandoned Luggage Detection System* with the example of silhouettes

In the implementation of this system, the distance signal features are kept in the text files and it is read every time the system is started.
If this classifier is run, it will give one of four possibilities of classes, they are:

- human
- human Group
- bag
- unknown.

If all the features have been calculated, the object will be classified in which class the object belongs by means of a decision tree. The decision tree is depicted in figure 34:

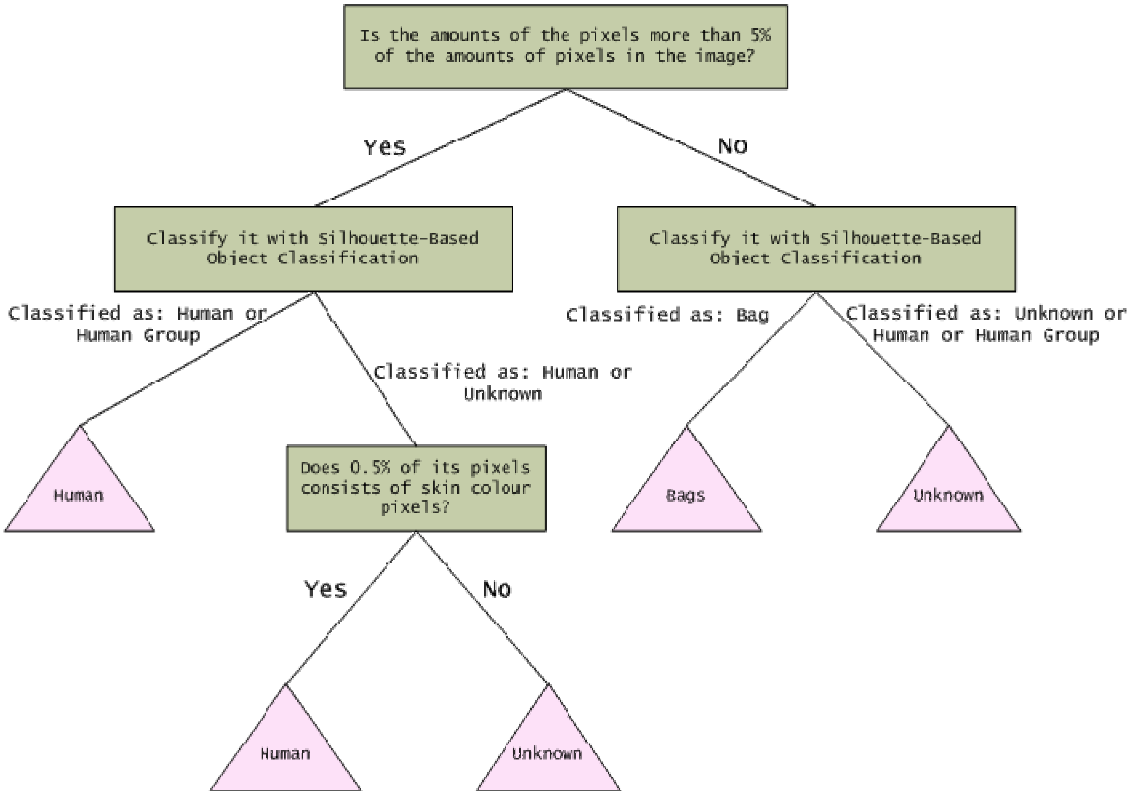

Is the amounts of the pixels more than 5% of the amounts of pixels in the image?
Yes
NO
Classify it with Silhouette-Based Object Classification
Classify it with Silhouette-Based Object Classification
Classified as: Human or Human Group
Classified as: Human or Unknown
Classified as: Bag
Classified as: Unknown or Human or Human Group
Human
Does 0.5% of its pixels consists of skin colour pixels?
Bags
Unknown
Yes
No
Human
Unknown

Figure 34: Decision tree to classify the object

6.2.4 Object Labeller

In the following section we will describe the implementation of the labelling process of the implemented system. In order to analyze the behaviour of the objects in the scene it is important to keep track of the object from the previous frame and current frame. For the implementation of the labelling process we have considered four scenarios:

- A new object enters the scene
- The object is from the previous frame
- Two or more objects from previous frame are merged in current frame
- An object from the previous frame is unmerged in the current frame.

6.2.4.1 The Blob from Previous Frame

For each of the scenario described above, a routine will be made. First the system will check if the blob in the current frame exist in the previous frame, if this is true the blob in the current frame will be assigned the ID in the previous frame. This is achieved by setting a threshold for the minimum threshold for the distance between the centre of mass point between blobs in previous frame and current frame (see figure 35).

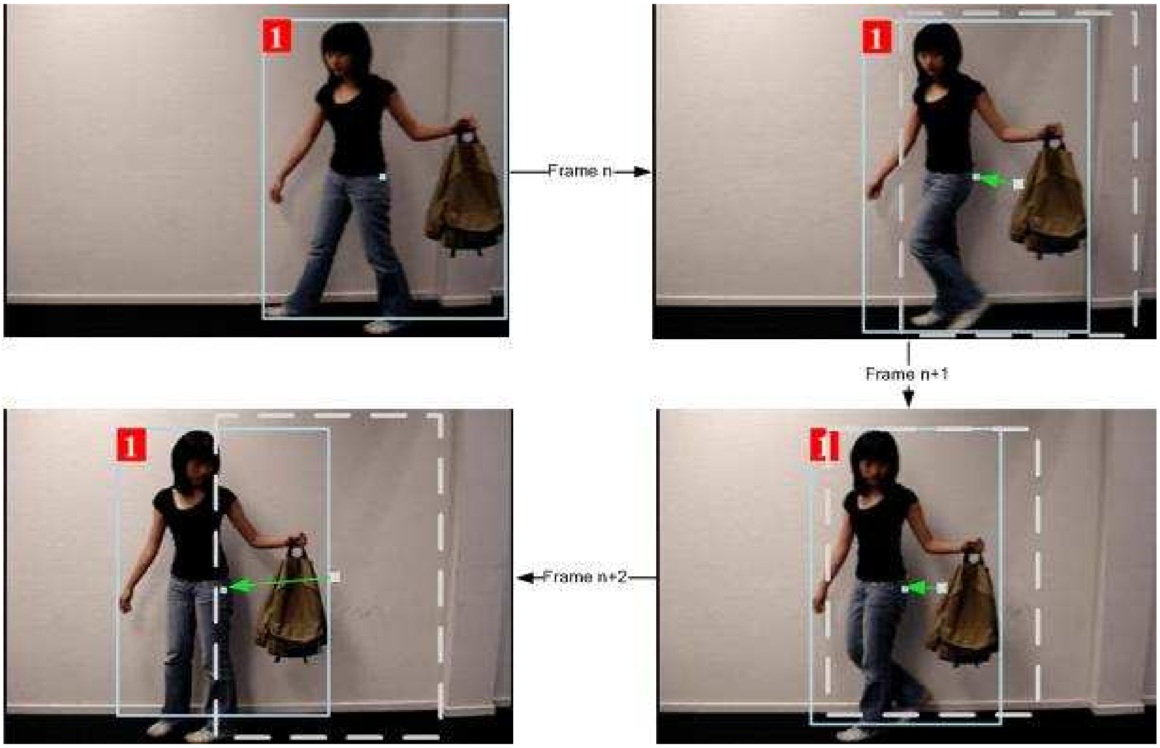


Figure 35: Blob 1 being tracked from frame to frame. They grey dotted bounding box illustrates the blob in the previous frame

6.2.4.2 Merged Blobs

The next scenario to check is when multiple blobs from the previous frame are merged into one blob in the current frame. From the blobs being merged, the ID of the largest blob will be used as root blob in the current frame. The rest of the blobs that are merged will be added to the root blob as child (see figure 36).

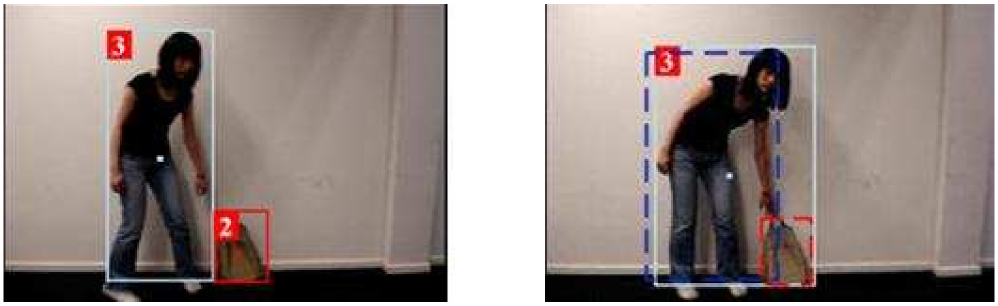


Figure 36: Blob 3 and 2 merged into root blob 3 with child blob 2. The dotted lines illustrates the bounding box of the previous frame

6.2.4.3 Unmerged Blobs

The unmerging process can be divided in two categories. The first category blobs without child are unmerged into multiple blobs. In the other case the blob has child and the child are unmerged and removed from the root node. When the unmerging happens without child, the unmerged blob will simply get a new ID assigned. An example of this is when a traveller enters the scene with a luggage, when the luggage is separated from the traveller a new ID will be assigned to the luggage. In the other case, where the blob being unmerged has a child the unmerged blobs in the current frame will tend to find the most likely child blob from the previous frame. Features such as centre of mass point colour, size and shape can be used to get the most likely blob from the previous frame (see figure 37).

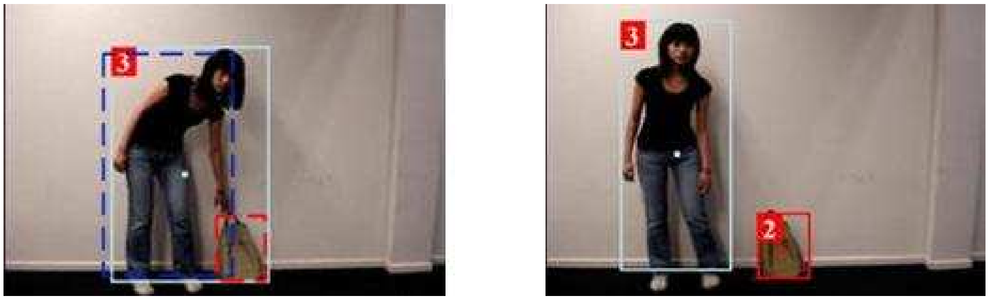


Figure 37: Blob 3 with child 2 unmerged into separate blobs 3 and 2.

6.2.4.4 New object enters the scene

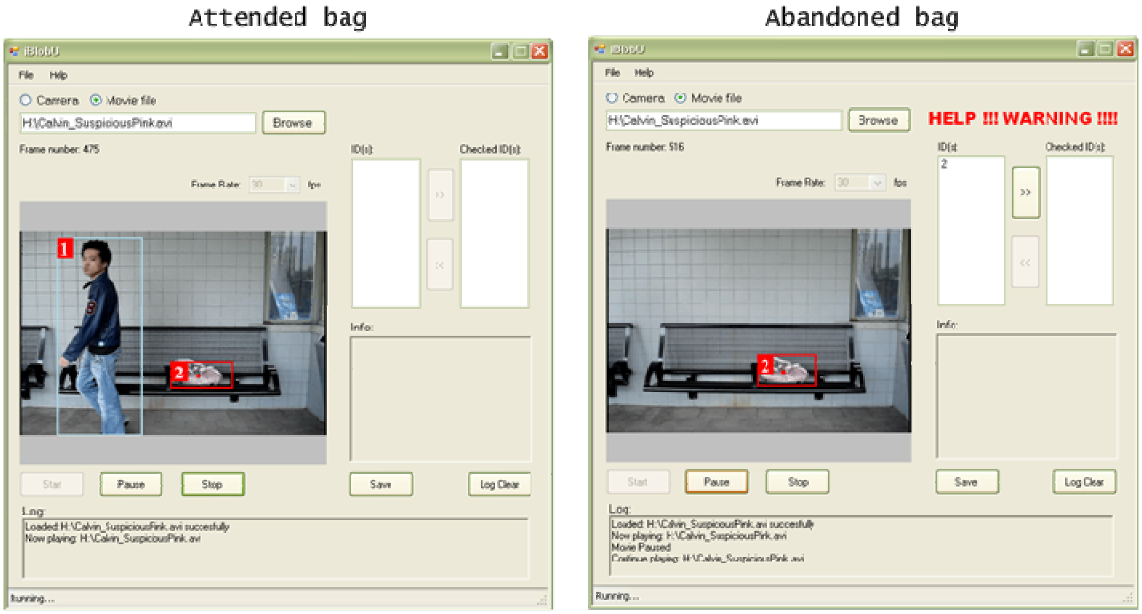
When none of the cases described above applies, the blob in the current frame is probably a new blob and a new ID is assigned to the blob.

6.2.5 Behaviour Detector

If the object has been detected and it can be tracked, then its behaviour can also be analyzed. In our system, the behaviour of the bags is detected to decide whether a bag is abandoned. There are two behaviours that a bag can have:

- Attended Bag
  A bag is detected as attended if there is a person in a radius of 250 pixels from the centre of the bag
- Abandoned Bag
  A bag is detected as abandoned if there is no person in a radius of 250 pixels from the centre of bag and the bag is not moving.

Warning is triggered when the behaviour of a bag in the movie is detected as abandoned. In figure 38, the screenshots of the *iBlobU Abandoned Luggage Detection System* when a bag is attended and a bag is abandoned will be displayed.



Attended bag

Frame number: 475

Frame Rate: 30 fps

Log:

```
Loaded H:\Calvin_SuspiciousPink.avi succesfully
Now playing: H:\Calvin_SuspiciousPink.avi
```

Running...

Abandoned bag

HELP !!! WARNING !!!!

Frame number: 516

Frame Rate: 30 fps

ID(s): 2

Log:

```
Loaded: H:\Calvin_SuspiciousPink.avi succesfully
Now playing: H:\Calvin_SuspiciousPink.avi
Movie Paused
Continue playing: H:\Calvin_SuspiciousPink.avi
```

Running...

Figure 38: An example of attended and abandoned bag

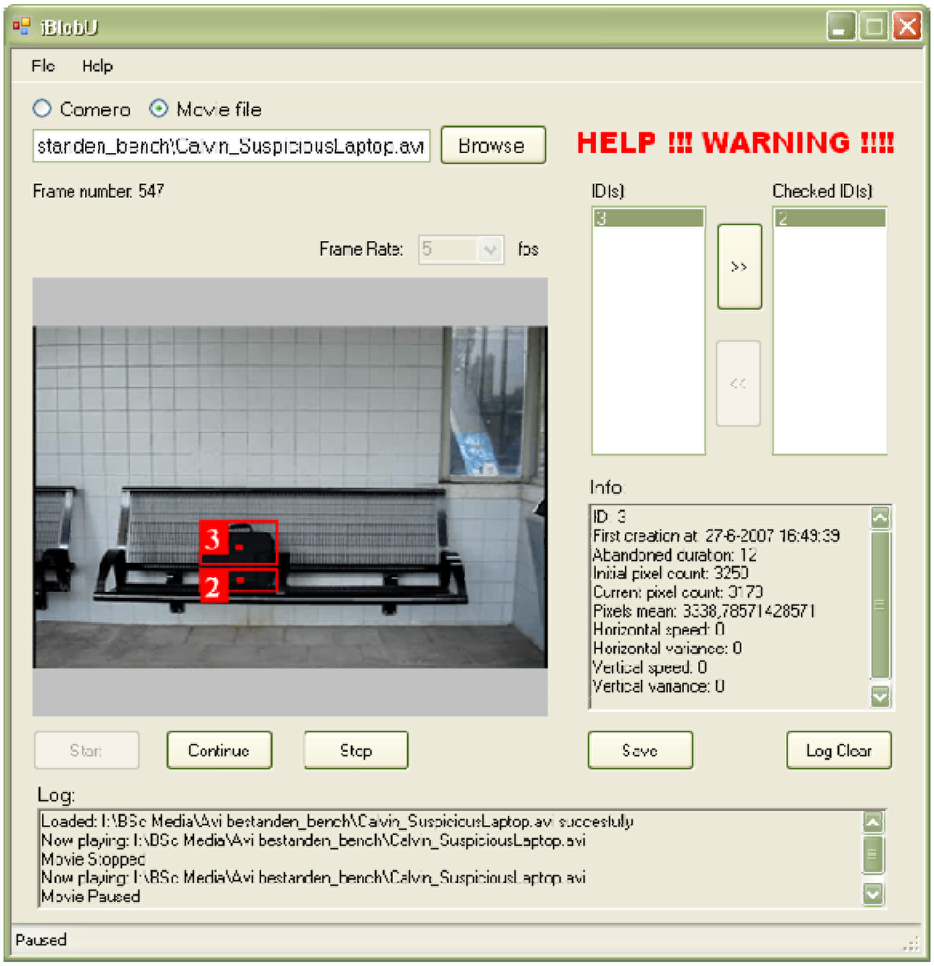
6.2.6 The *iBlobU* Graphical User Interfaces

As stated in the requirements, three graphical user interfaces will be made. Two GUIs will be made for the *iBlobU Abandoned Luggage Detection System* and one GUI for the *iBlobU Database*. In this section we will describe the implementation and usability considerations made during this project.

6.2.6.1 The *iBlobU Abandoned Detection System* GUI

First we will describe the GUI implementation of the supporting system that may be used by a security guard. Next we will describe the implementation of developer GUI.

In figure 39 a screenshot is provided for the *iBlobU System*. During the implementation of the GUI, various usability considerations have been made. We will give a short description for components used in this GUI.



standen_bench\Calvin_SuspiciousLaptop.avi

Frame number: 547

Frame Rate: 5 fps

ID(s): 3

Checked ID(s): 2

Info:

```
ID: 3
First creation at: 27-6-2007 16:49:39
Abandoned curaton: 12
Initial pixel count: 3250
Current pixel count: 3173
Pixels mean: 3338,78571428571
Horizontal speed: 0
Horizontal variance: 0
Vertical speed: 0
Vertical variance: 0
```

Log:

```
Loaded: I:\BSc Media\Avi bestanden_bench\Calvin_SuspiciousLaptop.avi succesfully
Now playing: I:\BSc Media\Avi bestanden_bench\Calvin_SuspiciousLaptop.avi
Movie Stopped
Now playing: I:\BSc Media\Avi bestanden_bench\Calvin_SuspiciousLaptop.avi
Movie Paused
```

Paused

Figure 39: The *iBlobU* Graphical User Interface

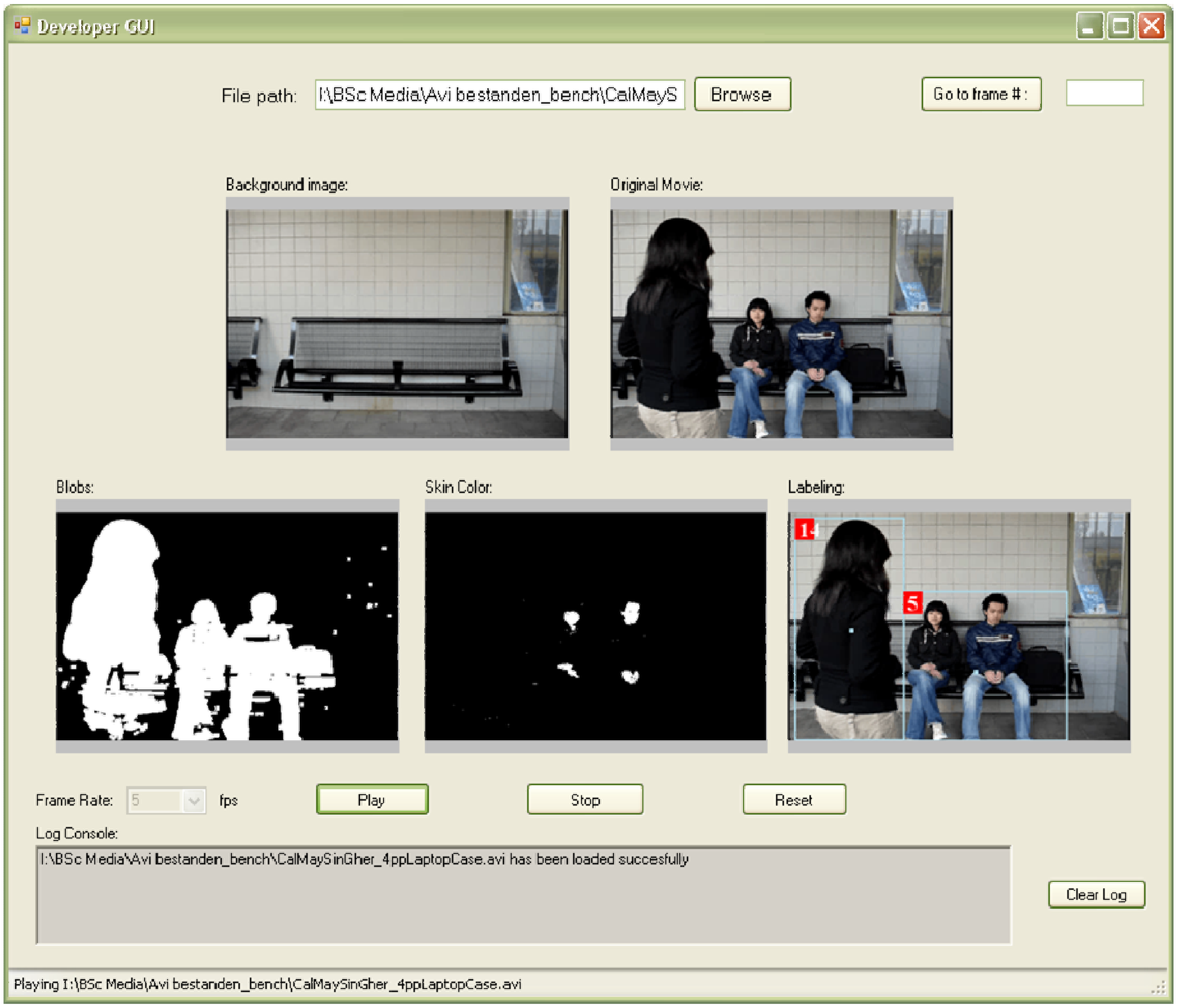
- **Input Source**
  For usability reasons it is conventional to provide the user with more than one solution to perform the same task. In order to select the input movie file that the user wants to analyze he or she can browse to the file either by pressing the browse button or via File>Open in the task menu bar.

- **Movie Controls**
  In order to prevent exception, all the movie controls are disabled until a movie file has been selected by the user. This assures that a null pointer exception is avoided when the movie controls are pressed while the user forgot to the select a source for the input. After a source is selected the user can use the movie control buttons below the picture box. Although the camera function is not implemented during this project, we can expect the system to disable all controls in the GUI except the start button when the camera is chosen as input source.

- **Warning Handling**
  In order to give users a clear separation between warning and general information, all the warning related components are grouped on the right side of the screen. The advantage is that related controls can be found easily by the user. When abandoned objects are detected visualization and audio will be triggered in order to warn the user

and catch their attention. One of the requirements of the user interface is that the user is able to control the triggered warnings. Therefore, two list box are provided where ID of abandoned object are listed. Using the buttons between the two list boxes the user can remove warnings that are checked or undo the removal when he or she accidentally added the warning to list of checked warnings. This control gives the user the ability to remove warnings in a non destructive way. Finally, detailed information of the detected warnings can be shown in the text box below by simply clicking on one of the IDs listed in either two list boxes.

In figure 40 a screenshot is provided for the *iBlobU* Developer GUI. As with the *iBlobU* GUI above, we will give a short description of usability consideration made during the implementation of the GUI.



I:\BSc Media\Avi bestanden_bench\CalMayS

I:\BSc Media\Avi bestanden_bench\CalMaySinGher_4ppLaptopCase.avi has been loaded succesfully

Playing I:\BSc Media\Avi bestanden_bench\CalMaySinGher_4ppLaptopCase.avi

Figure 40: The *iBlobU* Developer GUI

As we can see in figure 40, in comparison with the *iBlobU* GUI for the general user, the developer version consist of multiple movie frames. Each movie frame represents a different state during the *iBlobU System* processing. By visualizing the various state of the system processing, the developer GUI can give the developer detailed information from input to output. We will discuss some of the frames in the GUI shortly.

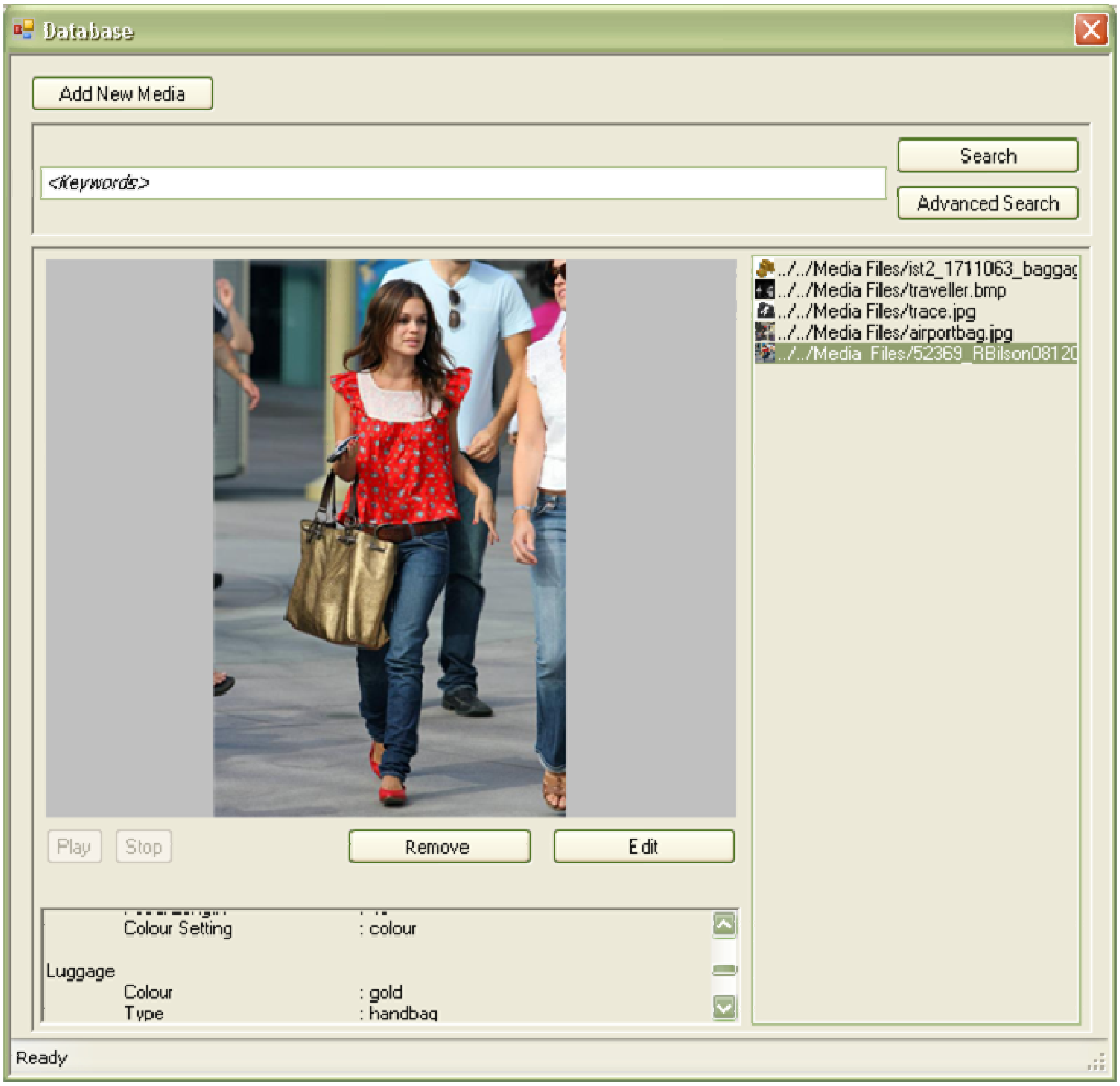
- **Background image**
  This image frame provides the developer the image of the initial frame of the movie being analyzed. Because background subtraction algorithm is used for the segmentation, it is convenient to save the initial background if the developer wants to perform manual comparison by eye during the system processing.

- **Blobs Movie Frame**
  The blobs movie frame provides the developer with the result from the segmentation process. This is one of the most important frame for the developer GUI. This is because the result of the segmentation is dependent for the rest of the following processes such as classification and labelling. By providing the developer a visualization of the segmentation output, the developer can verify if the labelling and classification process was performed correctly.

- **Labelling Movie Frame**
  The last movie frame in the developer GUI provides the developer with the visualization of the final output which will be showed to the end-user. In this frame the ID numbers are assigned to the detected blobs and the developer can verify f the system processes went correctly by comparing the segmentation movie frame with the labelling frame. Classification by the system can also be verified by the developer by checking the bounding box and comparing the labelled movie frame with the original frame.

6.2.6.2 The *iBlobU Database* GUI

The iBlobU Database GUI provides the developer an easy to use tool to manage media files in a database. By using the database the developer can effectively add new media files and search for specific media files that he or she is interested in. In this section we will describe implemented GUI functionalities.



../../Media Files/ist2_1711063_baggag
../../Media Files/traveller.bmp
../../Media Files/trace.jpg
../../Media Files/airportbag.jpg
../../Media Files/52369_RBilson08120

Colour Setting : colour

Luggage
Colour : gold
Type : handbag

Figure 41: The iBlobU Database GUI

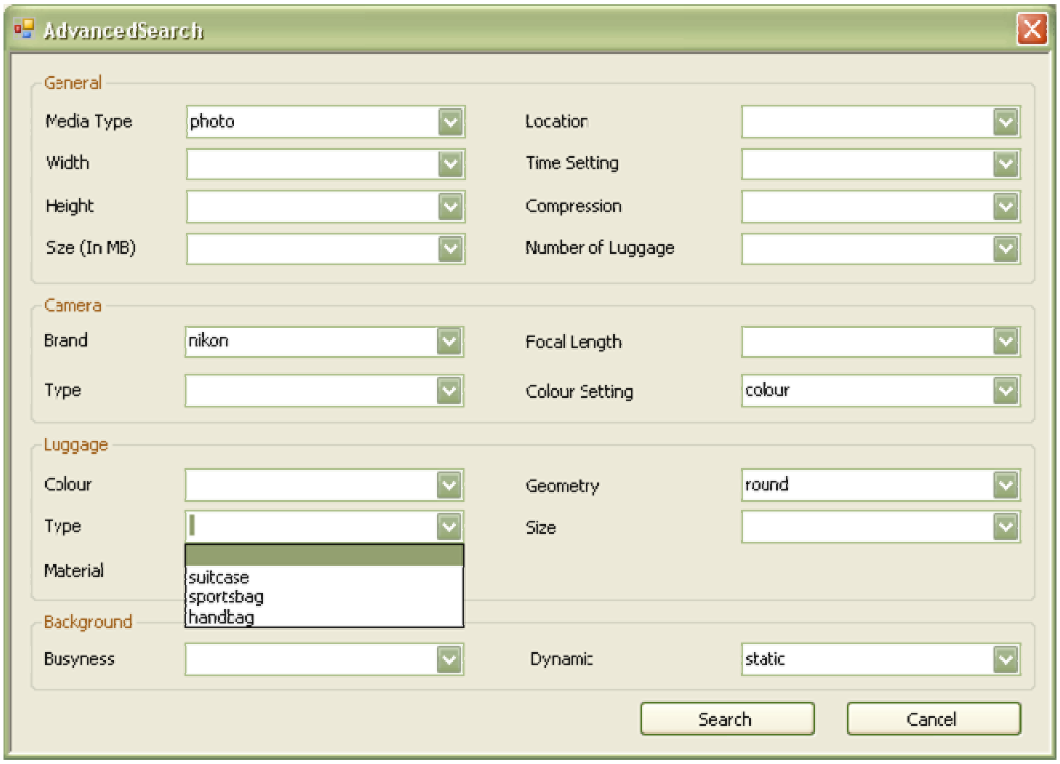
When the database system is started up, the developer is presented with the main screen (see figure 41). From this screen on the developer can add new media and search for media files in the database. The left side of the screen provides the developer a large view of the media files. By providing basic movie controls in the GUI the developer don't have to switch application in order to play the movie files. On the left side of the GUI a list is used to preview the result from a search. Icons are used in order to give the developer a preview of the file. In order to search for media files in the database, the user has two options, these are:

- **Search by keywords**

The first option to search for contents in the database is by using the search at the top of the GUI. By filling in key words, a query string to the database will be automatic generated and send to the database. The input of the user can be separated by spaces, dots or commas. The system will automatically cut the search strings in keywords and use it as search criteria's.

- **Advanced Search**

  The second option to search the contents in the database is to use the advanced search option. The advanced search can be accessed by pressing the advanced search button that will open a new window as depicted in figure 42.



Media Type: photo

Brand: nikon

Colour Setting: colour

Geometry: round

Dynamic: static

Figure 42: The iBlobU Database GUI, Advanced Search

The advanced search window provides the user with all the properties in which the user can search for. The advantage of the advanced search is that the user may discover properties of interest that he or she may have overlooked when the keyword interpreter was used instead.

Another important feature of the iBlobU database is to give the user a convenient way of adding new content to the database. When the Add New Media is pressed, the user will be presented with a new window as shown in figure 43 below. By choosing the media file to be added, the properties field will be activated and the user can fill in the required fields and add the information to the database by sending a new query containing the properties filled in by the user.

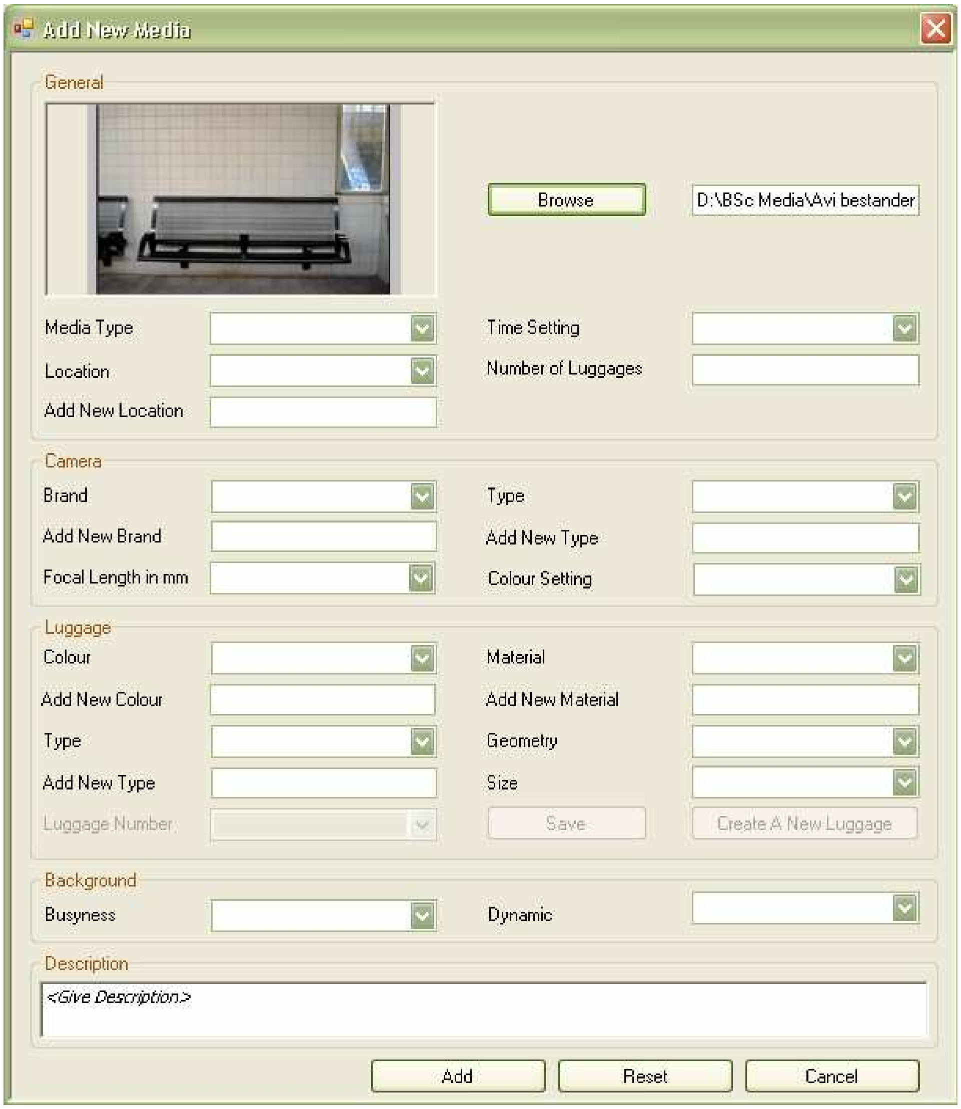


Figure 43: The iBlobU Database GUI Adding New Media

While the user can add new content to the database and search it. It is also important that the user can make changes to items that are already added to the database. For example, when a user added new content to the database and discovers that some properties were entered wrong. By choosing the edit option the main database GUI the edit window will be showed with the properties of the selected items. The items can then be changed by the user and n update query will be generated and send to the database to update his modified item.

7 Evaluation

"The greatest mistake is trying to be more agreeable than you can be."
Walter Bagehot

7.1 Test Methodology

Several aspects have been tested in order to conduct the progress of the system. The main goal of our system is to detect abandoned luggage and be able to classify between human and luggage objects. In this section we will first describe the procedure used for the evaluation of the system during the project. Next we will discuss the software and user test and finally the test results and recommendations will be given at the end of this chapter.

7.2 Procedure

As stated in the introduction, within the scope of this project several constraints have been specified in which the system will operate. During the design and implementation of the software, several scenarios have been made in which the system will operate. A complete list of scenario's can be found in Appendix B.4.5.1. To test the system, movie shoots have been made after the scenarios were defined. Because of the incremental approach of the implementation of the system, movies with same scenarios but different complexity have been recorded as the complexity of the system grows during the project. Not all the scenarios specified in Appendix B.4.5.1 have been used to test the system. Selections to the movies that satisfy the given constraint have been chosen with the main focus on bench scenes. The advantage of recording the scenarios with different complexity is that it gives us the opportunity to test the functionalities of the system during an early stage of the development of the software. This reduces the chances that problems will be found too late in the development process of the system and cause a ripple effect that affect other parts of the system when the system gets more complex.
The recorded movies can be categorized in the following two groups:

- movies recorded in a room under lab condition
- movies recorded in train station environment

The movies recorded in lab condition were shot in favourable conditions, the following constraints were applied:

- the room in which the movies are recorded is well lighted
- the light condition in the room remains constant while recording the movie

- the background colour is evenly distributed
- the contrast between the foreground and background is high enough to distinguish foreground and background colour
- maximum two persons in the scene
- only one luggage will be used
- the position of the camera is steadfast.

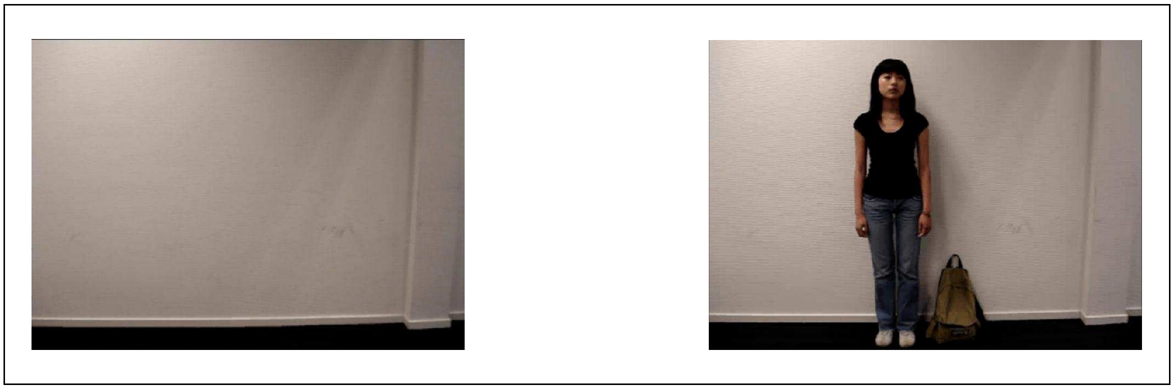


Figure 44: Snapshots of movie recorded with lab conditions

The constraints that are used for the movie shoots on Delft-Zuid train station can be found in chapter 4.3.

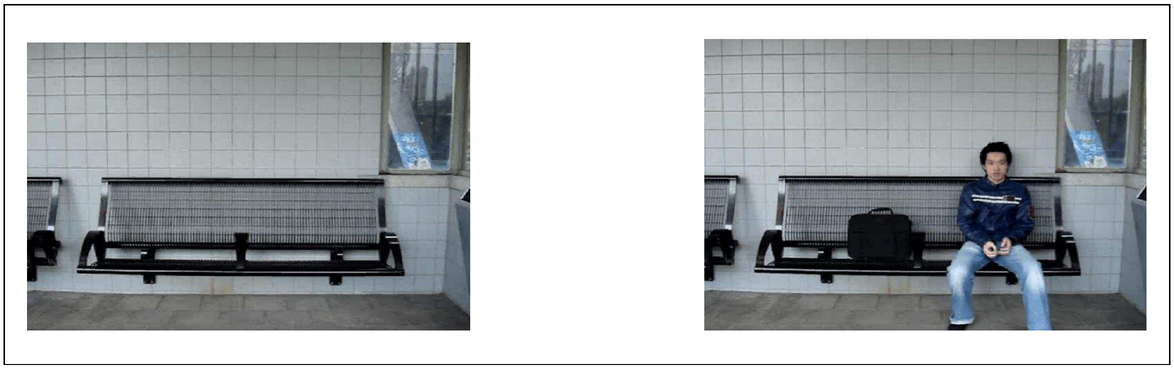
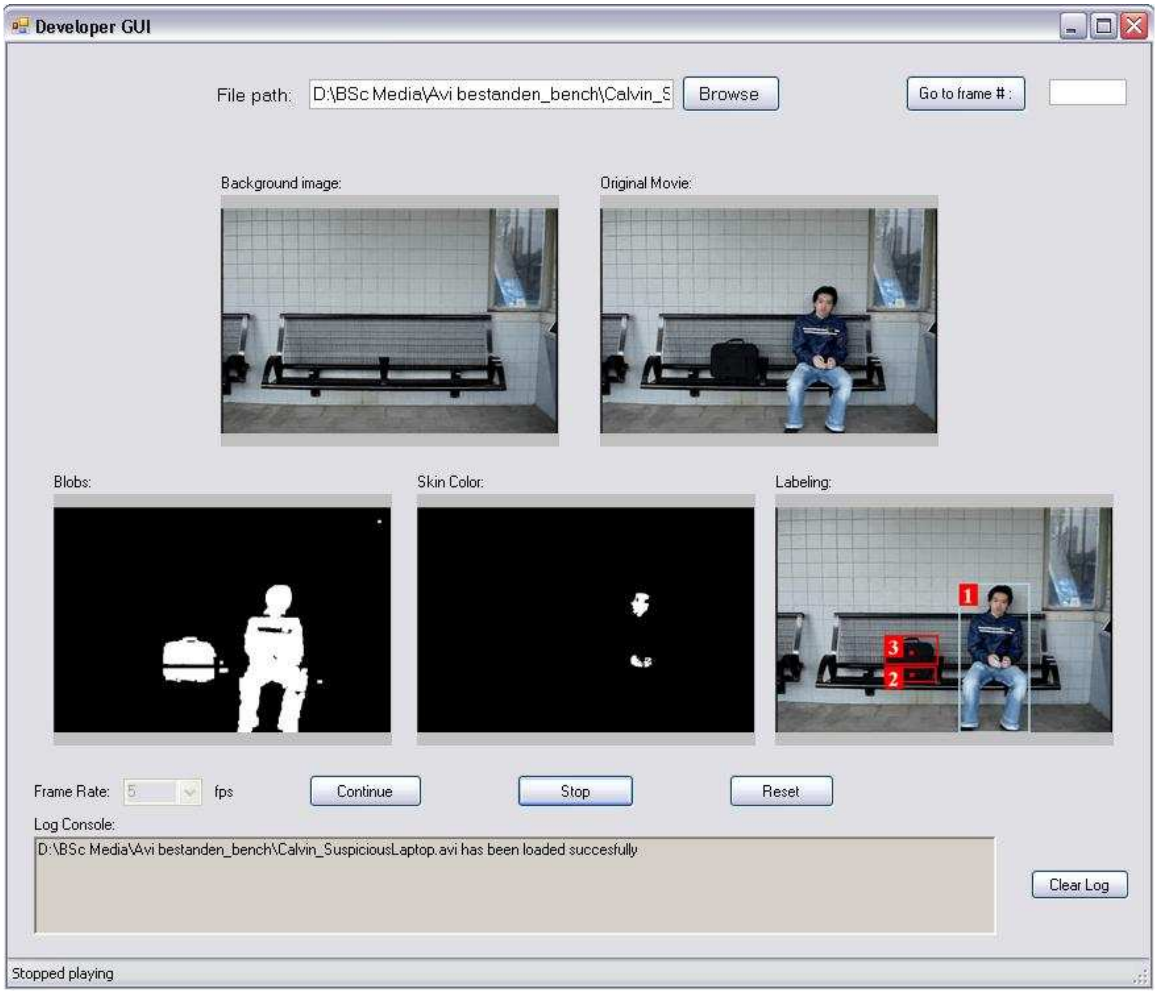


Figure 45: Snapshots of movie recorded on Delft-Zuid Train Station

7.3 Software Test

The implemented software consists of several core functionalities that are needed in order to achieve our goal of detecting abandoned luggage. To test the system, no test classes were made to perform software testing. Instead, movies with different scenarios were used to check whether the desired output is achieved. In order to follow the process that is performed to the input, visualization and console prints are used for the output of the process. These functionalities, which various states of the core processing was visualized, were implemented in the developer GUI (see figure 46). Besides testing the input of the system with the desired output, we also performed system performance tests. The performance of the system is measured in processing time. By calculating the difference of the timestamps which are placed before and after an operation, the processing time can be calculated. Since the system uses AVI movie files for input, it is important to test the influence of the frame rate of the movies on the accuracy and performance of the system. This is also a way to estimate the performance if a camera was used under similar conditions as in the recorded movies.



File path: D:\BSc Media\Avi bestanden_bench\Calvin_S

Frame Rate: 5 fps

D:\BSc Media\Avi bestanden_bench\Calvin_SuspiciousLaptop.avi has been loaded succesfully

Stopped playing

Figure 46: Snapshot of developer GUI visualizing classifier and labelling

7.4 Software Test Results

During the software tests, there were no problems encountered where the system resulted in a crash. Exceptions, which have been occurred during the test, have to be fixed immediately before the next test is started. In the rest of this section we will describe the result of the test for the core processes of the system.

- **Segmentation**
  From the test result we can conclude that this process is crucial for the final output of the system. This is because the quality of the segmentation affects the result of the blob creation, labelling and behaviour detection process. An example of a bad segmentation resulting in a wrong labelling is showed in figure 46 where one luggage is separated in two parts. Although this may not always lead to a problem for the abandoned luggage detection, it is not favourable. Also because the first hundred frames of a movie are used as samples for the background modelling process, the segmentation is dependent on the initial hundred frames of a movie and therefore

sensitive to objects that may occur during the training period. This can result in foreground pixels being classified in the background model that may cause a poor segmentation of the fore- and background. Despite of the constraints needed in which the implemented segmentation algorithm works correct, and the heavy calculations needed for the background modelling process during the first hundred frames, the result of the segmentation process is very promising compare to the naïve approach of background subtraction using only the initial frame.

- **Blob creator**
  The blob creator will analyze the image and create blobs from object of interests using the binary image that received from the segmentation process. For each group of pixels that found in the binary image, the blob creator will create a blob object including properties of the blob being created, such as the size and the centre point of mass. An initial test using the blob creator resulted in redundant blobs creation besides the object of interest. This was caused by noise that remained after binary operations during the segmentation process. To compensate this problem, an additional check has been added to the blob creator. By putting a proper threshold on the size of a blob, the blob creator was able to remove redundant blobs.

- **Classification**
  The classifier tries to classify the created blobs into the categories human, luggage or other. In order to visualize the classification, different colours have been assigned to the categories. Blobs that are classified as human will be marked with a blue bounding box and the blobs that are classified as luggage will be marked with a red bounding box. Blobs that are not classified as human or luggage will receive a grey bounding box. During the test the system was able to successfully classify human and luggage objects using contour shape and colour features. Although all the objects of interest were classified successfully, we discovered that misclassification occurred when parts of blobs were separated as result of a bad segmentation. An example is given in figure 47, we see here that the head of the person is separated for one frame which is caused by segmentation. This misclassification is however not surprisingly because the form and the size of the head have great resemblance with a luggage when a blob is created.

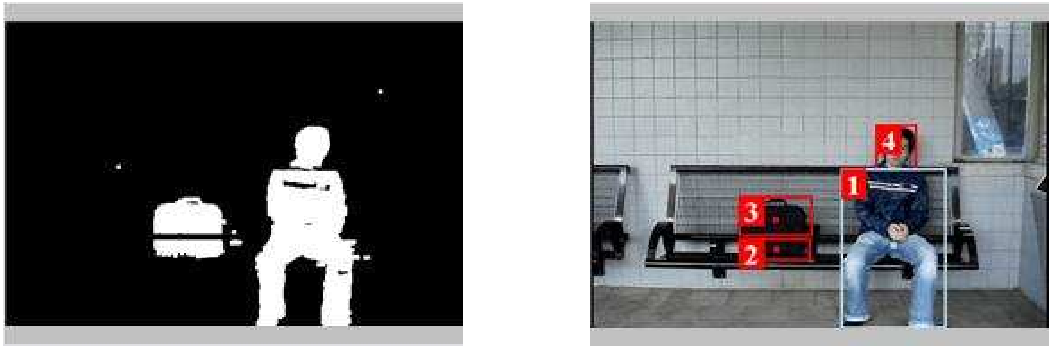


Figure 47: Misclassification of the head

- **Labelling**
  The labelling process has to assign each blob an ID and keep track of each blob in succeeding frames. During the test we have discovered that the unmerging part of the labelling process yielded difficulties with the assignment of the correct IDs when 3 or more blobs merged together. The merged children blobs can not be updated anymore while the properties may have changed drastically by the time that the blobs unmerge again. Also, because the blobs are tracked according to their centre of mass point

between previous and current frame, the frame rate affects the accuracy of the tracking process. A high frame rate may result in a better tracking of the blobs, but it will also result in low performance because of the number of frames processed per second. During the test we have discovered that a frame rate of 5 fps is a good trade-off between the accuracy of the labelling process and the system performance.

- **Behaviour detection**
  The behaviour detector was used to analyze the labelled blobs and trigger a warning when abandoned luggage is found. During the test with different kind of scenarios, the behaviour detector performed very well and was able to detect scenarios of abandoned luggage using simple properties such as the horizontal and vertical movement, mean size, variance and the distance to nearby objects.

7.5 User Test

Another aspect that we have aimed for is to provide the end-user a simple user interface with usability in mind. When designing the user interface we have tried to make the components of the GUI self explanatorily and resemble common windows controls that users are familiar with. To test the usability of the system, several users have tested the software. The computer experience of these testers is inexperienced in using computer. Prior the testing the goal and basic functionality of the system are told to the user. During the test we asked the user about events that may have occurred during the testing of the system such as graphical warnings and sound events. Finally, at the end of each test we asked the user if there are ambiguity while testing the system and what recommendation they have.

7.6 User Test Result

The most important feature of the GUI is to be able to warn the user when abandoned luggage is detected. After a basic introduction to the software, all the users that participated with the test successfully operated the system and were able to confirm detected abandoned luggage when warnings were triggered using graphics and audio. The user felt that the system was easy to use because the system does not have to be controlled actively until abandoned luggage is detected. In a real life situation this could decrease the workload of the security guards and they can concentrate on other tasks.

7.7 Test Conclusion

From the result discussed above we can conclude that the core requirements of the system are met. The system was able to distinguish the fore- and background. By analyzing the labelled blobs the system was able to classify human and luggage objects and trigger warnings to the user interface when abandoned luggage is detected. However, the result also showed that trade-off had to be made between performance and accuracy. For example, the background modelling process of the segmentation resulted in a much better segmentation than the default difference calculation implemented at the beginning, but on the other side it has caused noticeable decrease in performance. Also the unmerging part of the labelling processed caused wrong assignment of IDs. Although this is not favourable, it did not affect the detection of abandoned luggage.

7.8 Recommendations

In order to get a more robust system with higher performance, several aspects of the system can be improved. For instance, in order to get a more robust segmentation of the fore- and background, the calculated background model can be updated dynamically after the first hundred frames of training. Also, the labelling process has showed that the unmerging of blobs cannot rely on the centre of mass point alone. It is therefore recommended to add additional features to the blobs, this gives us better accuracy when finding the most likely blob when comparing blobs in the current and previous frame. Finally, the performance of the system can be improved in several ways. One way to improve the performance of the system is to use pointers in the implementation. The matrix operations in the ViewMatrix library and segmentation process showed significant performance increase when pointers were used instead of usual routines. Another way to improve the performance of the system is to process less frames per second. Although this may decrease the accuracy of the labelling technique used in this project, it can be compensated if a more robust object recognizer and classifier is being used.

8 Conclusion & Future Work

"People do not like to think. If one thinks, one must reach conclusions. Conclusions are not always pleasant." Helen Keller

This chapter will conclude this thesis. The goals set in the Introduction will be discussed in the first section, to see the progression of this project. It will then be followed by the Future Work, here the ideas that still can be developed in the future, will be presented. In the Concluding Remarks, general conclusion for the whole project will be drawn.

8.1. Goals

The goals that are defined in the beginning of this project will be reviewed one-by-one. In this manner, we can see how far these goals have been achieved.

1. *Designing and implementing an abandoned luggage detection system on the train or the area around it. This system will trigger a warning when an abandoned luggage is detected. This system will, however, have lots of constraints which are enumerated in the System Analysis chapter at section Scope of This Project. The goal is met when the system has been given a video from around the bench in the train station and it triggers a proper warning. See Appendix D for the script of this video.*

   Such a system has been designed and implemented. Two kinds of Graphical User Interfaces have been created for this system. The first one is a version which is intended for the developer. In this version, the user can see how the processing of the images is done and it also reflects in how the system came to the decision whether a luggage is abandoned. The second version is intended as a support system that gives a pre-alert warning for the user who analyzes videos. One example of such a user is a security guard.

   If the constraints are met, this implemented system will work properly as described in the System Requirements. However, there are aspects which still have to be improved; these are:

   - the processing time is still too long to be applied in a real time system
   - the system still does not have a proper connection with a camera, that makes it impossible right now to analyze real-time video from the camera. During the implementation phase, it appears to be more complex than expected. The real time camera requires a pre-calibration in order to work properly. This requirement is unfortunately not feasible for this project.

In spite of these deficiencies, this system works well with the movies taken around the bench in the train station where luggage is usually left behind. Therefore, the first goal has been achieved.

2. *Designing and implementing a database system where the information about the contents of the videos and photos can be managed.*

   This database system has been designed and implemented. The videos and photos are saved locally, since uploading these photos and videos to a server is not possible at this moment. This is because the server is not available and down- and upload time are slow, that can cause a bottleneck in the whole database system.

3. *Doing field research on the current video surveillance systems applied in the train and train stations.*

   During the project, we have visited the surveillance chamber at Amsterdam's train station and the central surveillance chamber at Utrecht. Every visit is documented as a report that can be found at Appendix F and G. The information has also been used in the Introduction in order to analyze the current situation.

4. *Produces videos and photos that illustrate the behaviour of the passenger with its luggage. These videos and photos can be useful as training data.*

   There have been two shooting sessions carried out at the Delft-Zuid Train Station. In these sessions, videos and photos have been taken. The photos usually display the way the passengers behave in the presence of their luggage. Pictures of abandoned luggage are also taken. As training data, the videos describe usual and unusual behaviour of passengers in the train station. The script of the videos can be found in Appendix D.

8.2. Future Work

The possibilities for improvement of the abandoned luggage detection system in the future are described below.

1. *Updating the background model dynamically*

   In our implemented abandoned luggage detection system, the modelling of the background is only done once, when the system has been started. But the modelling can also be done continuously when the detection process has been activated. This can be implemented by applying multiple layers of background model. The algorithm and technique of updating the background model are presented in [6].

2. *More hypothesis for labelling*

   The labelling in our implemented system works well for tracking non-moving and slow moving objects which is already sufficient for detecting abandoned luggage. But to make the tracking more precise, more hypotheses on tracking the object can be added. Examples of these hypotheses are:

   - taking the colour of the objects in consideration
   - taking the shape of the objects in consideration.

3. *A fluent connection with the camera*

The processing of the images can also be applied for camera images in order to do real-time detection. By implementing a connection with the camera that is able to capture images fluently, this can be achieved.

4. *Tracking the luggage that belongs to a passenger*

   By linking a luggage with a passenger, the other kinds of suspicious behaviours of the passenger can be detected, such as exchanged luggage, stolen luggage, etc. However this will demand that the background segmentation technique has to be more advanced.

5. *Creating a network of cameras*

   Deploying a network of cameras can be handy when the tracking area becomes wider. This can be done by letting the cameras communicate with each other.

8.3 Concluding Remarks

In the first section of this chapter, the goals of this project have been analyzed. It can be concluded that the goals are mostly achieved. Aspects that should have been improved during this project are the slow performance of the implemented abandoned luggage detection system and the connection of the camera to the system which should have been implemented.

In the course of this project, we found out that the detection of the abandoned luggage is not a trivial issue to solve. Implementing such a system gives most of the time a dilemma between improving the robustness of the system what usually causes the performance of the system to be slow, or improving the speed of the processing time what usually reduces the robustness of the system or will increase the cost of the system.

As a result of this project, lots of documentation and resources like videos and photos have been produced. These products will hopefully be able to contribute to the research on abandoned baggage detection in the future when the technologies that are used nowadays will become more advanced.

References

[1] Dedeoglu, Y., Toreyin, B.U., Gudukbay, U., Cetin, A.E. (2006). *Silhouette-Based Method for Object Classification and Human Action Recognition in Video*, Springer-Verlag Berlin Heidelberg, 2006.

[2] Guler, S., & Farrow, M.K. (2006). Abandoned Object Detection in Crowded Places. *Proceeding 9th IEEE International Workshop on PETS*, 99-106, 2006.

[3] Hazlewood, P. (2006, November 2). *Britain becoming a 'surveillance society'*, Retrieved June 11, 2007, from http://www.ioltechnology.co.za/article_page.php?iSectionId=2885&iArticleId=3517941

[4] Kim, K., Chalidabhongse, T.H., Harwood, D., & Davis, L. (2005). *Background Modelling and Subtraction by Codebook Construction*, Computer Vision Lab, University of Maryland, 2005.

[5] Kim, K., Chalidabhongse, T.H., Harwood, D., & Davis, L. (2005). *Real-time foreground-background segmentation using codebook model*, Elsevier Ltd, 2005.

[6] Kim, K., Harwood, D., Davis, L.S., (2005). *Background Updating for Visual Surveillance*, Springer-Verlag Berlin Heidelberg, 2005.

[7] Kirkup, M., & Carrigan M. (n.d.). *Video surveillance research in retailing: ethical issues*, Retrieved June 11, 2007, from http://www.emeraldinsight.com/Insight/ViewContentServlet?Filename=Published/EmeraldFullTextArticle/Articles/0890281103.html

[8] Li, L., Luo, R., Huang, W., & Eng, H. (2006). Context-Controlled Adaptive Background Subtraction. *Proceeding 9th IEEE International Workshop on PETS*, 31-38, 2006.

[9] Lv, F., Song, X., Wu, B., Singh, V.K., & Nevatia, R. (2006). Left-Luggage Detection using Bayesian Inference. *Proceeding 9th IEEE International Workshop on PETS*, 83-90, 2006.

[10] Manning. S. (2007, February 26). *Cameras taught to watch and learn*, Retrieved June 11, 2007, from

http://www.ioltechnology.co.za/article_page.php?iSectionId=2885&iArticleId=3702503

[11] Peer, P., Kovac, J., Solina, F. 2003. *Human skin colour clustering for face detection*. In submitted to EUROCON 2003 – International Conference on Computer as a Tool.

[12] Smith, K., Quelhas, P., & Gatica-Perez, D. (2006). Detecting Abandoned Luggage Items in a Public Space. *Proceeding 9th IEEE International Workshop on PETS*, 75-82, 2006.

[13] Thirde, D., Li, L., & Ferryman, J (2006). Overview of the PETS2006 Challenge. *Proceeding 9th IEEE International Workshop on PETS*, 47-50, 2006.

[14] Wu, B., & Nevatia, R. (2006). Tracking of Multiple, Partially Occluded Humans based on Static Body part Detection. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition- Volume 1(CVPR'06)*, 951-958, 2006.

Appendix A

Proposal

A.1 Introduction

Nowadays with the increasing risk of terrorism, the detection of abandoned luggage in public transportation and its surroundings is a crucial task. The difficulty of the current way of luggage detection represents the risk which the passenger has to deal with everyday. The current procedure of handling abandoned luggage is time consuming and most of the time it is just a false alarm. Its way of detecting suspicious abandoned luggage is also inefficient. An alarm is only triggered when someone notices suspicious luggage or when security guards observe the suspicious luggage manually by means of camera captures.

In this document we propose a system that will recognize and track luggage. Furthermore this system will analyse traveller's behaviour with his or her belongings and will trigger warning signals to the user when suspicious luggage or behaviour have been detected. With help of our system, the detection of suspicious abandoned luggage will be more efficient and accurate. The time-consuming actions which are done to handle false alarms will also be lowered. Therefore it will increase the sense of safety for every passenger.

In the beginning of this document, we will give a brief problem description followed by the requirement of the system. In this requirement section, we will explain which features does our system has and some necessary constraints to make it work properly. Next, our approach in producing this system will be discussed. Finally a summary will be given.

A.2 Problem Description

"Developing a user-friendly system which is able to recognize and track luggage in the public transportation and its area. When a suspicious left behind luggage is detected, the system must trigger warning signal to the user"

A.3 Requirements

In this section, we will present all the functionalities and constraints of our system.

A.3.1 Functional Requirements

A.3.1.1 General Functions

- The system must be able to recognize single or multiple luggage
- The system must be able to track single or multiple luggage
- The system must be able to work real-time within 5 seconds tolerated delay
- The system must be able to trigger warning when it detects suspicious abandoned luggage
- The system must also have a database which contains images and movie clips that describe various scenarios
- The system must also give the opportunity to add or remove the data in the database
- The calibration must be done manually

A.3.1.2 Database

- Database must consist of images and movie clips
- Given some specific query, the user must be able to search images and videos in the database
- Images must be able to be added to or removed from the database

A.3.1.3 User Interface

- The user interface must be able to visualize the result of the recognition and tracking done by the system
- The user interface must support manual calibration
- The user interface must trigger a warning signal (audio or/and visual) if it detects suspicious abandoned luggage
- The user must be able to indicate the false positive warnings and undo the warning signals

A.3.1.4 Implementation

- The system is implemented in the C# programming language
- The system is using OpenCV to support image processing operations
- The database is implemented in MySQL

A.3.2 Constraints

A.3.2.1 Hardware

- The computer which will run the system has to meet the following conditions:
  - Operating System: Windows XP
  - Memory: minimum 1 GB RAM
  - Processor: minimum 1.8 GHz Intel or an equivalent
  - Microsoft .NET 2.0 Framework
  - Camera colour: 32-bit RGB
  - Camera frame rates: minimum 30 fps
  - Camera resolution: minimum 640 x 480 pixels

A.3.2.2 Environment Condition

- In open areas, e.g. train station, the camera must be placed:
  - at the corner where there the obstruction is least visible
  - between 3 - 5 meters above the platform
  - facing down (approximately 30 degrees from the horizontal)

- In closed areas, e.g. in the train, the camera must be placed:
  - in every row of the seats and at the end of the aisle
  - between 2 - 3 meters above the floor
  - facing down (approximately 30 degrees from the horizontal)

- The luggage must be fully visible to the camera
- The fluctuation of the lighting condition must be minimized
- The input image must not be under- or overexposed
- The colour between the luggage and the background must be distinguishable
- The luggage must be one of the following items:
  - briefcase
  - suitcase
  - backpack
  - guitar case

A.3.2.3 Quality Assurance

To assure the quality of the system, the system has to suffice the following categories:

- Performance

  The system works efficiently, so that it is able to work in real-time within the tolerated delay

- Accuracy

  The system recognizes 80% from the visible luggage and an error-rate of 10% false positives of all abandoned luggage

- Robustness

  The system recognizes invalid input of the camera and can handle a degree of noise caused by the environment

- Maintainability

  The system is extendable upon more advanced features and has complete documentations for both users and developers

A.4 Approach to Requirements

The requirements mentioned above need some approaches to help realizing the system. These approaches will be discussed in this section. There are three prototype phases which describe the procedure of the whole project.

A.4.1 First Prototype

This first prototype should be the simplest of all prototype phases. The system will take only one input image and it has to be able to recognize the luggage shown in the image. In this first phase, the object and the background are static. A database file is going to be needed to save all images and to train the system. At the end of this phase, the system should be able to classify the luggage.

A.4.2 Second Prototype

In the second prototype phase, the input will be a sequence of images and thus not a single image anymore. The objects, the luggage and the travellers, are now allowed to be dynamic but still the background should be static. By giving the possibility to have dynamic objects, the database of images and video clips will be extended automatically in comparison to the database from the first prototype. The output from this phase will be a system which able to distinguish travellers and luggage in sequent. The luggage is also coupled to the owner of that luggage. The final product will be build after this phase is completed. The approaches to the final product will be handled in following passage.

A.4.3 Final Product

The final phase gives more flexibility for the system input. The objects and background are both allowed to be dynamic and a sequence of images is still required for the input. If there is any suspicious left luggage detected, then the system will trigger a warning signal to particular authorities. Additionally, the system has to be able to recognize the traveller who abandoned the luggage.

A.5 Planning

The whole project has to be completed within 8 weeks and thus the estimated time needed for each phase has to be planned. The planning is as follows:

- Analysis phase: 1 week
- Implementation phase
    - Prototype 1: 2 week
    - Prototype 2: 2 week
    - Final Product: 2 weeks
- Final evaluation phase: 1 week

During each implementation phase, testing and evaluation will be performed. Furthermore, the feedback will be given to the achieved results frequently.

A.6 Summary

The proposed system tracks abandoned luggage in public transportation or its surrounding. By analysing the traveller's behaviour with their luggage, this system will be able to trigger an alarm when a luggage is abandoned. Although this system is not going to perform perfectly, it will help to reduce the crime and terrorism or eventually it is possible to track down to whom the luggage belongs in crowded places e.g. train station and airport.

Appendix B

Requirement Analysis Document

B.1 Introduction

Many public or open areas are facilitated with cameras to monitor the security of that space. Sometimes it is needed to have an area watched from different angles to make sure that there is nothing wrong going on with the security. This situation has to be done by human since that is the only solution at this moment. But still, there are some shortcomings in the existing way and security guards want to operate this by automation. Here, a system will be proposed to give a solution to this problem.

The goal of this report is to give all parties clarity about the system that will be built. This way there will be no misunderstanding between all parties and the programmers know what has to be implemented.

In this report the current situation will be described shortly followed by the concept of the system. Then the proof of concept will be given in the next section. After this the future vision will be discussed and in the last section the summary will be given.

B.2 Current Situation

Nowadays with the increasing risk of terrorism, the detection of abandoned luggage in public transportation and its surroundings is a crucial task. The current procedure of handling this abandoned luggage problem is time consuming and inefficient. The screens from all cameras that hang in certain areas have to be continuously monitored as in real time. It is almost impossible to have the security guards observing all the actions on the screen from all camera views at most of the time. That is why many false alarms occur. An alarm is only triggered when someone notices unusual phenomena such as suspicious traveller's behaviour or abandoned luggage.

B.3 Concept

Analyzing the current situation, a system is needed to automate the video surveillance system. An ideal system will be able to track people and their luggage. When the system has the ability to do this, the behaviour of the traveller and his or her luggage can be analyzed. By comparing real data with normal and suspicious behaviour, suspicious behaviour (e.g.) abandoned luggage will also be able to be detected.

By means of this kind of system, the public transportation provider and the authority have the capability to control the security and give the travellers a saver feeling in the public transportation area in an efficient and effective way.

In the following section, a concept of automated video surveillance system is proposed.

B.3.1 Proposed design

The goal of the proposed system is to be able to detect abandoned luggage. Given an input it has to trigger warning if abandoned luggage is detected within the input. To achieve this result the proposed system will consist of various components and subsystems, which are:

- input is a real time stream from camera
- image enhancement of the received input
- foreground subtraction
- people and Luggage recognition of the foreground using classifier (expert system)
- people and luggage tracker
- analyzing behaviour, behaviour recognition using classifier (expert system)
- triggering system
- database system containing still images and videos which are used to train the classifier
- matching user interface for the various actors

This is depicted in figure B.1.

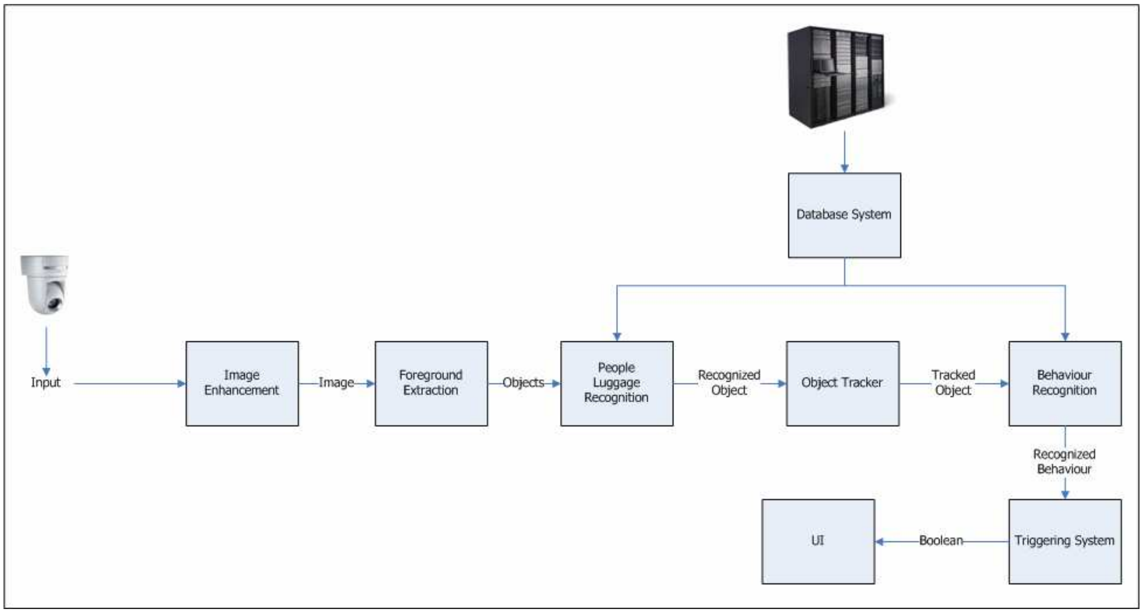


Figure B.1: General flow diagram

Input

The camera captures the images in real-time, these images will be fed to the system for further processing. In order to achieve an ideal result several constraints have to be met by the camera. First, the camera has to be mounted on places where the view is clear and where potential abandoned luggage is likely to be found. Furthermore, the place where the camera is placed should be adequately exposed to light.

Image Enhancement

The aim of image enhancement is to improve the quality of the image before further processing. This can be achieved by the following techniques: blurring filter, RGB normalization, changing colour space, etc. The output image should have less noise and reduce the influence of illumination.

Foreground Extraction

In order to extract the region of interest, the foreground should be extracted from the image. This way the background does not have to be considered anymore in further processing. The result of the foreground subtraction will affect the accuracy of the object recognition. To distinguish the foreground from the background, several known techniques exist. The most known technique used for foreground extraction is by subtracting the initial state image from the current image.

People and Luggage recognition

To understand the semantics of the extracted foreground, this process has to distinguish the various kinds of objects that may occur in the scene. These objects include people, luggage and other moving objects. However, before the recognition can take place, we have to examine the input first and then extract the features that are useful. Examples of these features are shapes, texture and colour. Using the generated rules from the knowledge based system, it will be possible to classify the various objects.

Object Tracker

The next step in the system is to keep track of the recognized objects in the scene. Because the proposed system has to process images received in real time it is important that objects in the scene are being tracked over time. The information in each processed frame will be used in the next step to classify different behaviours. To track objects techniques such as optical flow and hidden Markov chains can be used.

Behaviour recognition

Since the objects are recognized and its movement are analyzed with respect to its surroundings, the behaviour can be determined. Depending on the actions of the objects that are being tracked we can classify the recognized behaviour into two classes: normal and suspicious behaviour. Some techniques that can be used to realize this are pattern recognition techniques and neural networks. The features that are used to classify the behaviours are for example velocity, motion history gradient and distance to other objects.

Database

To train classifiers, the object and behaviour recognition, a database will be built to automate the process of collecting and retrieving training data. This database system will also have a User Interface to visualize the data and give a user-friendly way to add new data to the database.

Warning Triggering

When behaviour is classified as suspicious, in this case it implies abandoned luggage, the system will trigger a warning. A warning can be visual, audio or a combination of both.

User Interface

Different User Interfaces are designed for specific users. These interfaces are adjusted to the needs of the user. For example the User Interface for the developer will contain detailed information of the various processes in the system, while the interface for the end-user focus on the visualization and the warnings triggered by the system. For the user interface that will be used by the end-user (the security guard) it will contain a feature to control the warning signal and tell the system whether it should not trigger a warning signal.

B.3.2 Feasibility

Looking at the techniques and hardware needed to realize the proposed concept given above, it is possible to build this system. But there are some points which make it difficult to apply such system in the near future. These are:

- the quality of the camera which are used nowadays to oversee the area is not suitable to be used as an input for such system
- the computer vision and machine learning processes requires a lot of processing power
- luggage can have unpredictable appearance, this makes it very difficult to recognize all luggage

In order to provide an idea of the proposed concept a simplified version will be used as the proof of concept.

B.4 Proof of Concept

Within the framework of this project a simplified system of the concept will be built. But this system will be built around constraints which are needed, given the duration of the project.

The system will have two versions, the first one is a developer version, which will have a Graphical User Interface where the user can see the processing which is carried out by the system. This version of the system is used to analyze the behaviour of the system, which will be utilized to improve the performance of the system in the later development. This version can also be used by the security guard who supervises the public transportation area by means of the video captures. In this case, the security guard can also let the system know when it detects a false positive.

The second version is a rip-off version of the first version. In this version, Graphical User Interface is not provided. The user will know when abandoned luggage is detected by perceiving visual or audio warning. This kind of system can be employed in public transportation area to warn the travellers if they have forgotten to bring their luggage when they left the corresponding area. Therefore, this version will be named as traveller version.

In this section the following structure will be used. First, a general description of the implemented system and its constraints will be given which will be followed by the functional and non-functional requirement of the implemented system. In the next section, the pseudo-constraints will be explained. At last, the technical design of the implemented system will be discussed in the System Model section.

B.4.1 Implementation of the Proof of Concept

Techniques needed to implement each part of this system are chosen based on the feasibility of the techniques in the duration of this project. Figure B.2 visualizes the flow of the implemented system.

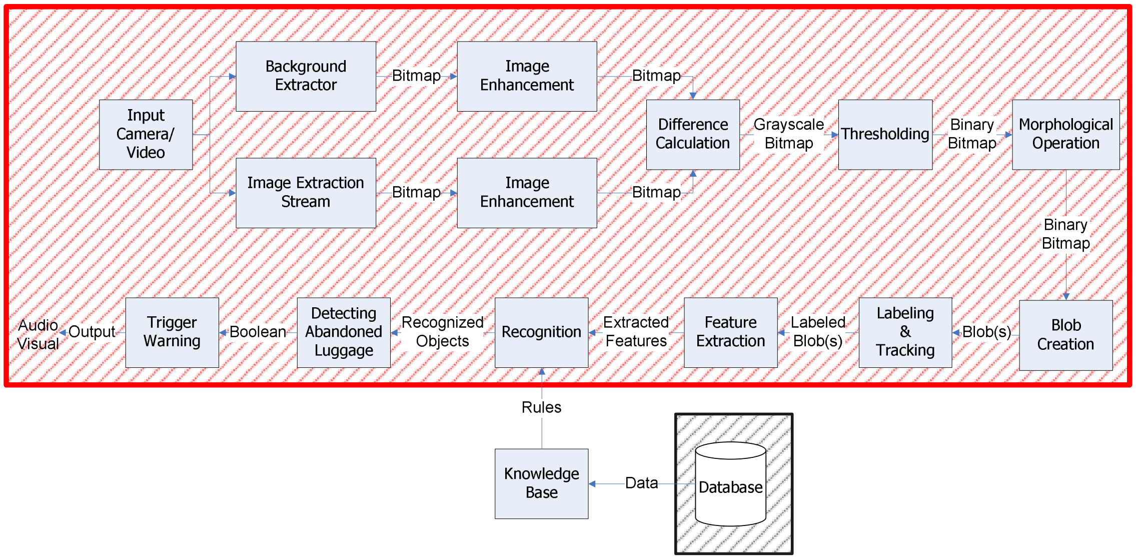


Figure B.2 The flow of the implemented system

The parts of the system that will be implemented are marked red and black. Although the black marked part of the system will be implemented, it will not be part of the abandoned luggage detection system, it will be used as a stand alone application for searching and adding training data in the database. The database system will not be discussed in this documentation, but it will be provided separately. The red marked parts is the actual system that will detect abandoned luggage, it contains the parts as discussed in the previous section where a description is given for a general concept for an abandoned luggage detection system. This system will be implemented based on blob extraction techniques.

Further, this system is built around the following constraints:

- input of the system is always an avi video file
- background is clear from objects and is only one evenly distributed colour
- light condition remains constant during the shot of the movie
- contrasts between foreground and background are high
- Luggage and traveller are not occluding each other when luggage is put next to the traveller
- only one luggage will be abandoned
- Distance between camera and object/background is between 2 and 2.5 meter. Frontal shot
- in the scenes there will be one person with or without a bag
- the initial state is free of foreground objects and used as background
- all frames will be processed by the system
- when a person walks out of the screen, and he or she comes back he or she will not be recognized as the same person
- warning will only be triggered when luggage is being abandoned
- 3 to 10 seconds allowed till warning is triggered when luggage is being abandoned

B.4.2 Functional Requirements

As mentioned in the previous section, there are three users that are going to use the abandoned luggage detection system, namely the security guard, developer and traveller. The security guard usually sits behind the screen watching the video which is captured real-time by the cameras. The developer is the one which has task to improve the system and the traveller is people who use the public transportation whose area is surveyed by the camera. The security guard and developer will use the developer version of the system, whereas the traveller will be provided with the rip-off of the developer version (traveller version).

Although there are 2 versions of the system which are going to be made, these versions share the same general functionalities, which are:

1. the system must be able to detect abandoned luggage
2. the system must provide a manual calibration which is used to find good values on processing the video images

The developer version will provide these specific functionalities:

1. the system must give a visual or audio warning to the user when abandoned luggage is detected by the system
2. the system must show the result of video processing and the reasoning of the triggered warning
3. the user must be able to report when a false negative is detected

The traveller version will provide this specific functionality:

1. the system must give a visual or audio warning to the user when abandoned luggage is detected by the system

B.4.3 Non-Functional Requirements

B.4.3.1 User Interface and Human Factors

1. The user interface can be used by people who have little knowledge of computer s
2. The audio or visual warning can be perceived without much effort

B.4.3.2 Documentation

A user manual will be made to describe all the features of the system and it has to meet these conditions:

1. a user manual has to be written that is understandable for all users
2. a user manual has to contain information about the system and how to operate the system

B.4.3.3 Hardware Considerations

The computer that will run the system has to meet the following requirements:

1. Operating System: Windows XP
2. Memory: minimum 1 GB RAM
3. Processor: minimum 1.8 GHz Intel or an equivalent
4. Microsoft .NET 2.0 Framework
5. When a video files are analyzed, the computer has to have AVI codec in order to view the video file
6. The camera which is used to take the frame image must meet the following conditions:
    a. colour: 32-bit RGB

b. frame rate: minimum 30 fps
c. resolution: minimum 640 x 480 pixels

B.4.3.4 Performance Characteristic

The object recognizing process must be performed as rapidly as required by the user or necessitated by the process being controlled, which is in real-time with 5 seconds tolerance delay. The quality of performance depends on the situation of the area in the scene:

1. the best performance will be achieved up to 95% from a scene with less than equal 4 objects
2. the best performance will be achieved up to 75% from a scene with more than equal 5 objects (or crowded area)

B.4.3.5 Error Handling and Extreme Conditions

1. The system must be robust
2. The system must activate the alarm when video input is not valid

B.4.3.6 System Interfacing

1. The system uses an existing face tracking system

B.4.4 Pseudo-Constraints

The system has to run on a Windows XP Operating System. The system is implemented in C# programming language with a .NET library for the image processing. The MySQL.net connector will be used to communicate with the database in C#. When video has to be analyzed, it has to be compressed in AVI.

B.4.5 System Models

B.4.5.1 Scenarios

Scenario: The system gives a warning
Actors
Bob: Security guard
Flow of Events

1. The system gets an image input from camera or video files
2. The system analyzes the images
3. The system detects an abandoned luggage
4. The system gives an audio or visual warning
5. Bob observes the alert and checks whether the detected abandoned luggage is suspicious

Scenario: The user turns off a warning which is given by the system
Actors
Bob: Security guard
Flow of Events

1. The system detects an abandoned luggage
2. The system gives an audio or visual warning

3. Bob observes the alert and checks whether the detected abandoned luggage is suspicious
4. Bob realizes that the alert is harmless and turns the warning off

Scenario: The user calibrates the camera
Actors
Bob: Security guard
Flow of Events

1. Bob starts the system and realizes that the video capture is not suitable to be processed
2. Bob chooses calibration function
3. Bob changes some variables which are provided by the user interface
4. The system adjusts the variables in the system by the variables which are given by Bob
5. The system shows the video capture with adjusted variables
6. Bob confirms to adjust these variables

Scenario: The system gives a warning
Actors
Alice: Traveller
Flow of Events

1. Alice sits in one of the bench (e.g. in the train or at the station) and put her luggage next to her
2. Alice left the bench without her luggage
3. The system detects an abandoned luggage
4. The system gives an audio or visual warning
5. Alice reacts to the system and realizes that she forget to bring her luggage

B.4.5.2 Use Case Model

Use case: Receiving a warning
Actor
Security Guard
Entry-condition
The system has detected an abandoned luggage
Flow of events

1. The security guard receives an alert from the system
2. The security guard searches for the abandoned luggage id on the screen
3. The security guard checks to whom this luggage belongs and where he or she is on the screen

Exit-condition
The security guard has checked the harmfulness of the warning

Use case: Turning of a warning
Actor
Security Guard
Entry-condition
The system has detected an abandoned luggage
Flow of events

1. The security guard receives an alert from the system

2. The security guard searches for the abandoned luggage id on the screen
3. The security guard checks to whom this luggage belongs and where he or she is on the screen
4. The security guard sees that it is a false alarm
5. The security guard presses on the Stop Warning button to turn off the warning

Exit-condition

The security guard has turned off the warning

Use case: Calibrating the video image

Actor

Security Guard

Entry-condition

The system has been started

Flow of events

1. The security guard chooses calibration function
2. The system returns a user interface where the variables can be adjusted
3. The security guard adjusts the variables
4. The system shows the video image with the adjusted variables
5. The security guard confirms the adjusted variables
6. The system adjusts the variable that used in the system with the given variables

Exit-condition

The system variables are adjusted

Use case: Detecting the abandoned luggage

Actor

Traveller

Entry-condition

The system has been started and calibrated

Flow of events

1. The traveller sits on bench whose area is surveyed by camera and put the luggage next to him
2. The traveller left the bench without the luggage
3. The system gives a audio or visual warning
4. The traveller came back to the bench and bring the luggage away
5. The system stops giving audio or visual warning

Exit-condition

None

B.4.6 Object Model

B.4.6.1 Data Dictionary

Abandoned

When luggage is unattended by the owner for a long period of time

Attended

Luggage that is near the traveller

Blob

Groups of connected foreground pixels labelled by the tracking estimate which overlaps

Calibration

Adjusting the intensity of the picture, in order to make it feasible to be processed

Luggage

Object(s) e.g. bag, box, suitcase, etc… a traveller takes along

Picture

A still image or a capture in a video

Recognizing

To give a semantic meaning to a group of pixels

Silhouette

A group of pixels which represents the contour of the object

Tracking

Following the object of interest

Traveller

Everyone who travels with the train

User

Everyone who uses the abandoned luggage detection system

Video

A sequence of still images representing scenes in motion

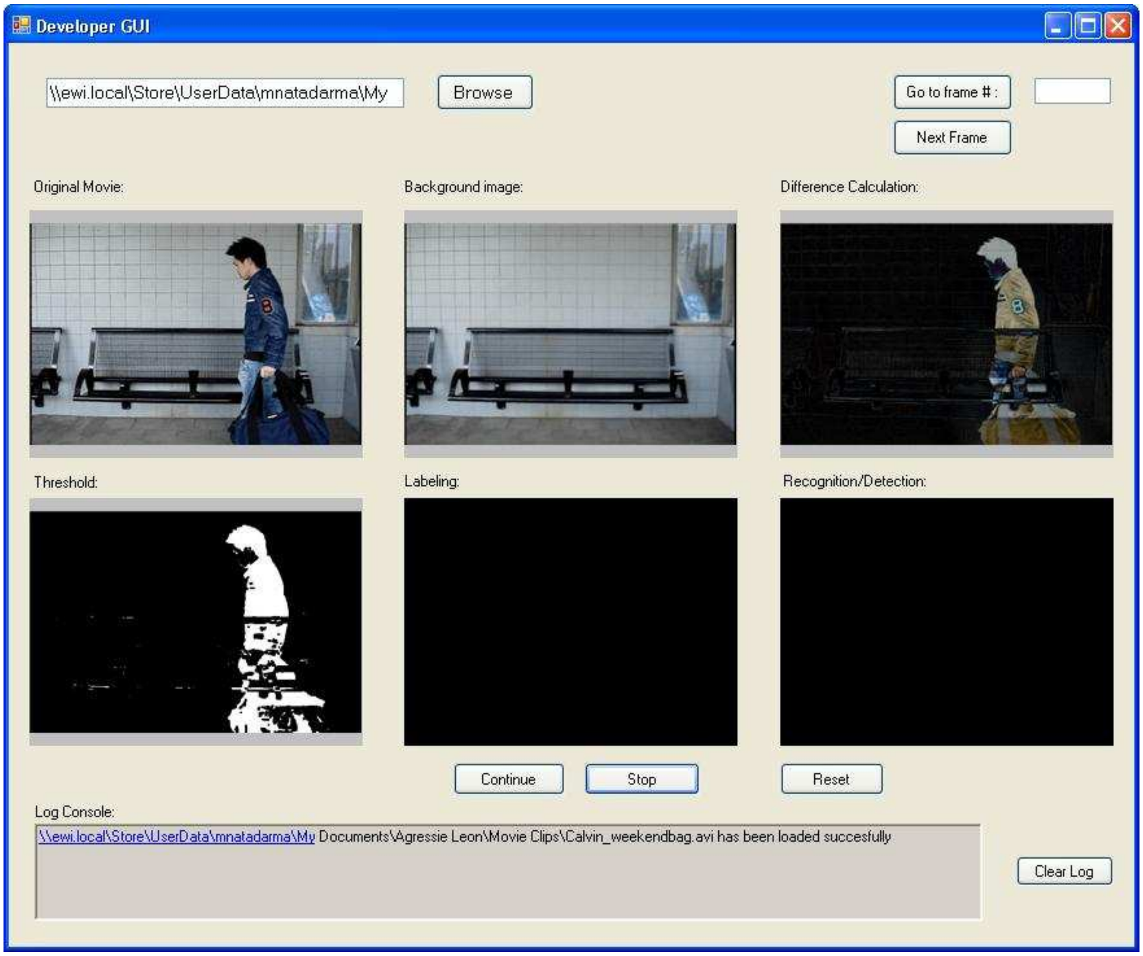
Warning

Visual and auditive warning, to alert the user on a suspicious event

Unattended

When there is a (small) distance between the traveller and his luggage for a short period of time

B.4.6.2 User Interface



\\ewi.local\Store\UserData\mnatadarma\My

\\ewi.local\Store\UserData\mnatadarma\My Documents\Agressie Leon\Movie Clips\Calvin_weekendbag.avi has been loaded succesfully

Figure B.3: The developer GUI

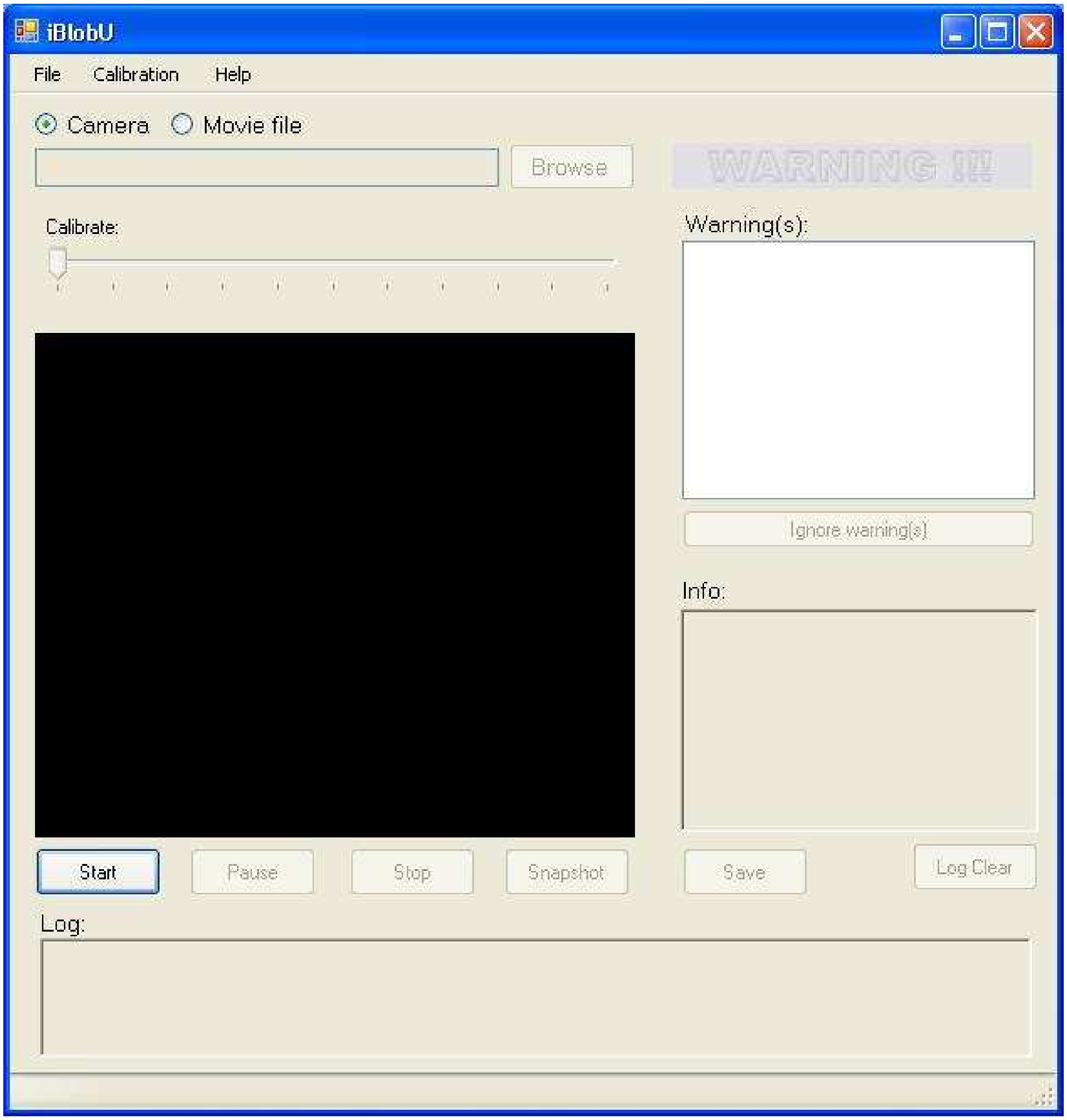


Figure B.4: The iBlobU GUI

B.5 Future Vision

While the concept of the proposed system may be really helpful to improve the level of safety and comfort, there are still improvements that are possible in the future. Some of the thinkable improvements are:

- **Network of cameras**
  Instead of using single input from the camera, it is possible to use a network of cameras. This will reduce the number of false positives. By having multiple views of the various object problems that can occur with only one camera such as occlusion and overlapping objects can be tackled. If the cameras can communicate with each other it is very helpful to track the moving objects in bigger areas.

- **Long period tracking**
  To track a person for a longer period, it is convenient for the system to know who this person is. For this an identical face tracking is important. This way a person can be identified by the system.

- **Central alarming system**
  When abandoned luggage is detected, in the current system warnings are analyzed manually by the end-user. It should be a ubiquitous system, this way the user will not notice the presence of the system. When assistance of authorities is needed, the system will communicate with a central alarming system and warn them to take action immediately.

B.6 Summary

In this document we have proposed a concept for a system for detecting abandoned luggage. The proposed concept will help to increase the effectiveness to detect abandoned luggage. To prove the concept, a simpler system will be implemented in this project. Although the proof of concept does not include all the functionality of the system proposed in the concept, it will help to analyse the video images and detect abandoned luggage in particular scenarios.

Appendix

C

Database Design

C.1 Database Tables layout

For each sample image or movie a media record will be made in the database. The attributes used for the representation can be found in the tables below.

- **Media table:** this table is the main table and contains the general information of the image or movie clip. Further detailed information of the camera, luggage and background are stored in the other tables
- **Camera table:** this table contains general information of the camera, and the used settings for a particular shot
- **Luggage table:** this table contains the general information of a luggage
- **Background table:** this table contains what the background setting was for a particular shot

C.1.1 Media table

The media table has the following fields:

- Id
- Type
- Width
- Height
- Size
- Location
- Time setting
- Date
- Compression
- Camera id
- Luggage id
- Background id
- n Luggage
- Filename
- File path

Description:

- Type: photo or video
- Width and height: describes the dimension of the media
- Size: in bytes
- Location: where the shot was taken
- Time and date: when the shot was taken
- Compression: jpeg, avi, mpeg etc.
- Camera id: id of an camera to get more detailed information of type and settings
- Background id: id of link to detailed information of the used background settings
- n Luggages: number of luggages in the shot
- Filename: filename
- Filepath: path location of where the file is stored

C.1.2 Camera table

The camera table has the following fields:

- Id
- Brand
- Type
- Focal length
- Colour setting

Description:

- Brand: producer of the camera
- Type: type of camera
- Focal length: focal length setting of camera
- Colour setting: black/white, colour etc.

C.1.3 Luggage table

The luggage table has the following fields:

- Id
- Colour
- Type
- Material

Description:

- Colour: main (dominating) colour of the luggage
- Type: suitcase, briefcase, guitar case, backpack etc.

C.1.4 Background table

The background table has the following fields:

- Id
- Busyness
- Dynamic

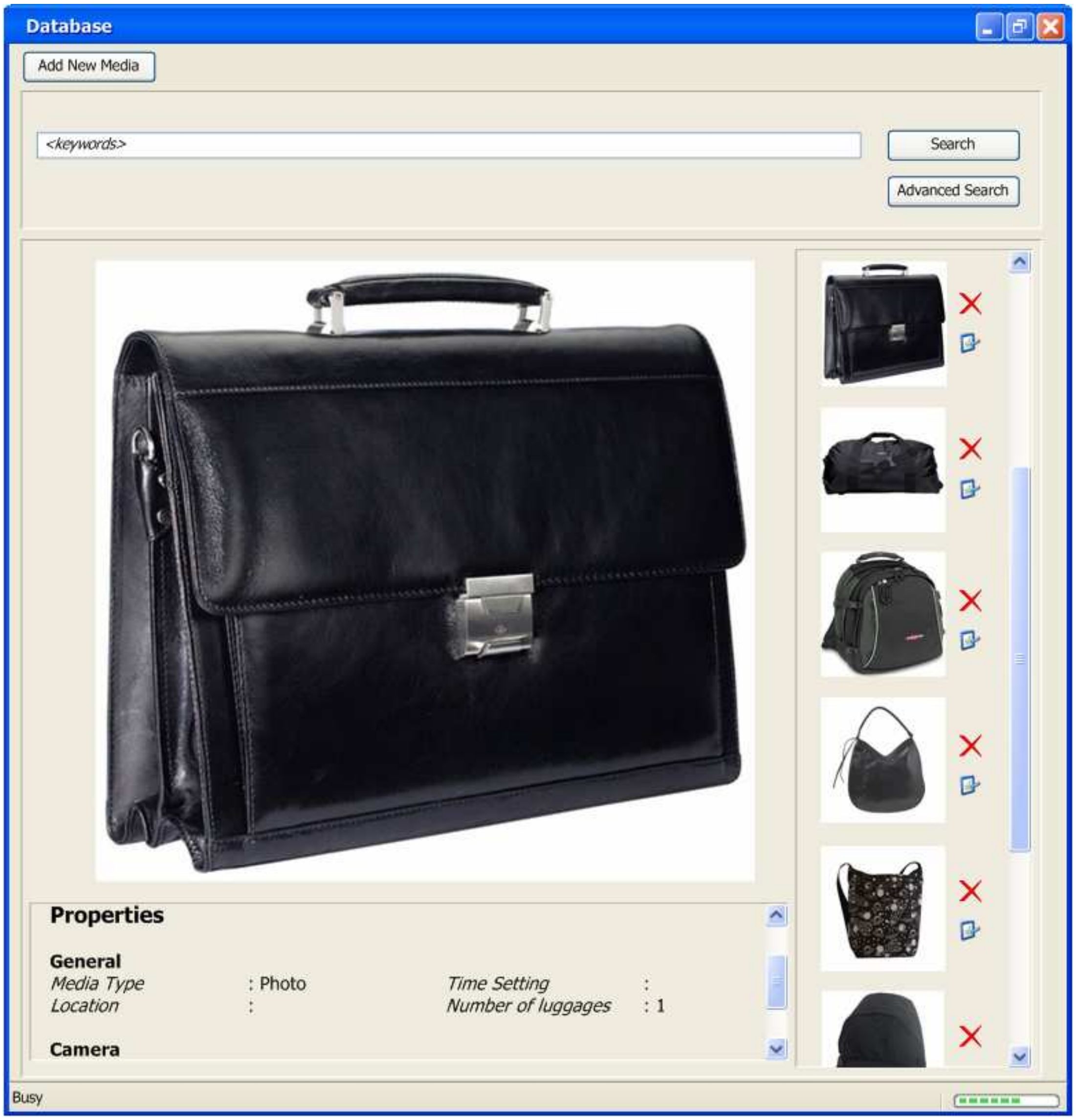
Description:

- Busyness: occupation of the scene, defined with scales
- Dynamic: moving background or static?

C.2 Database GUI

The database has a main screen as shown in figure C.1. In the main screen, it is possible to add, remove, edit or search a media file.

In the upper part the user can search for a file with keywords. In the lower part of the screen the search results are displayed as thumbnails at the right side. At the left side the picture will be shown when selected, the properties of the file is given beneath the picture.



Properties

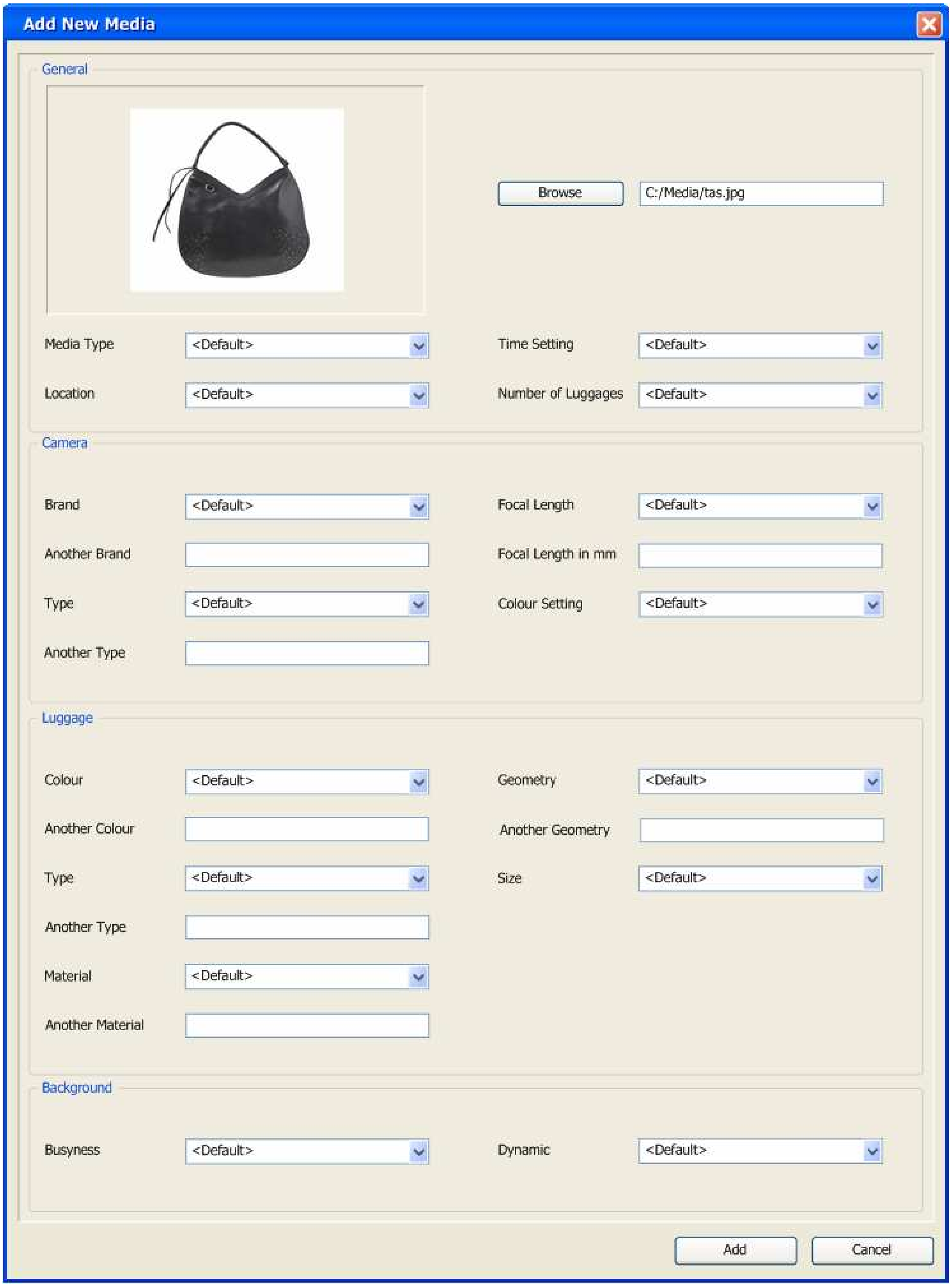
General

Media Type	: Photo	*Time Setting*	:
Location	:	*Number of luggages*	: 1

Camera

Figure C.1: Database main GUI

When the user presses the “Add New Media” button the screen in figure C.2 will pop-up. With this screen, the user is able to add a new file to the database. The user also has to complete the form in order to add an image or video.



Add New Media

General

Browse C:/Media/tas.jpg

Media Type	<Default>	Time Setting	<Default>
Location	<Default>	Number of Luggages	<Default>

Camera

Brand	<Default>	Focal Length	<Default>
Another Brand		Focal Length in mm	
Type	<Default>	Colour Setting	<Default>
Another Type			

Luggage

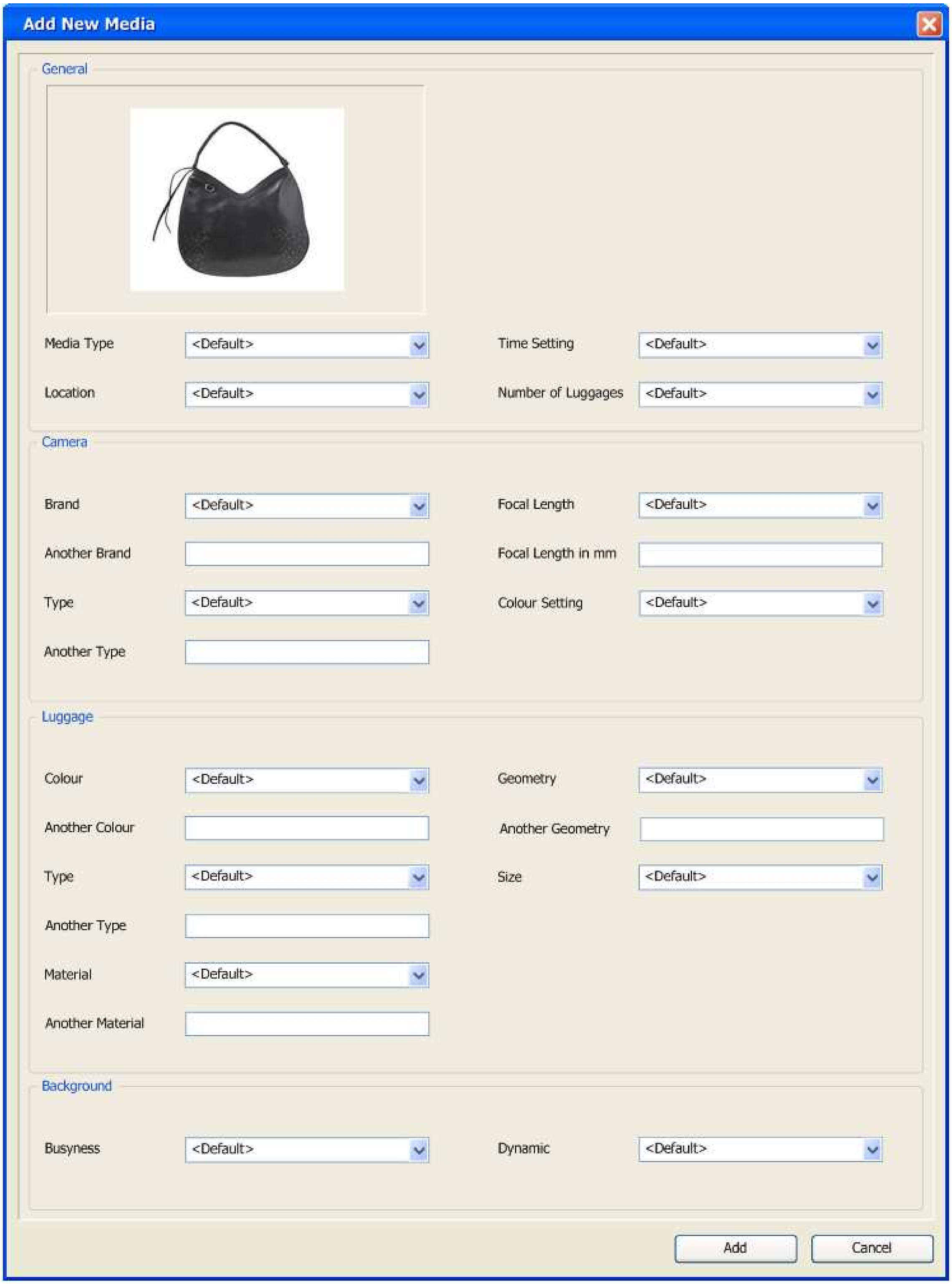
Colour	<Default>	Geometry	<Default>
Another Colour		Another Geometry	
Type	<Default>	Size	<Default>
Another Type			
Material	<Default>		
Another Material			

Background

Busyness	<Default>	Dynamic	<Default>

Figure C.2: Add new media GUI

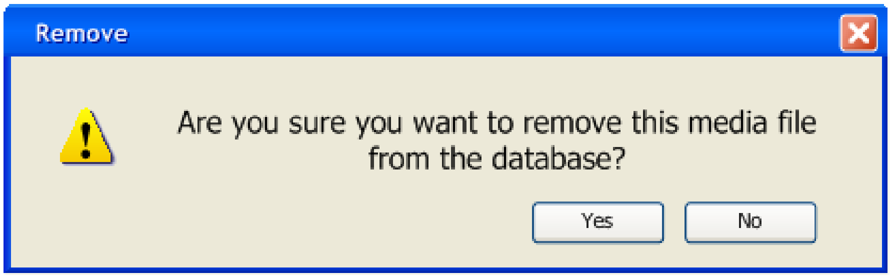
By displaying the search results there will be two buttons, namely: a remove and edit button. When the edit button is pressed, the screen as shown in figure C.3 will be shown. The user is now able to edit the information of the picture or movie.



Media Type	<Default>	Time Setting	<Default>
Location	<Default>	Number of Luggages	<Default>
Brand	<Default>	Focal Length	<Default>
Another Brand		Focal Length in mm	
Type	<Default>	Colour Setting	<Default>
Another Type			
Colour	<Default>	Geometry	<Default>
Another Colour		Another Geometry	
Type	<Default>	Size	<Default>
Another Type			
Material	<Default>		
Another Material			
Busyness	<Default>	Dynamic	<Default>

Figure C.3: Edit GUI

If the remove button is pressed the interface in figure C.4 will be shown.

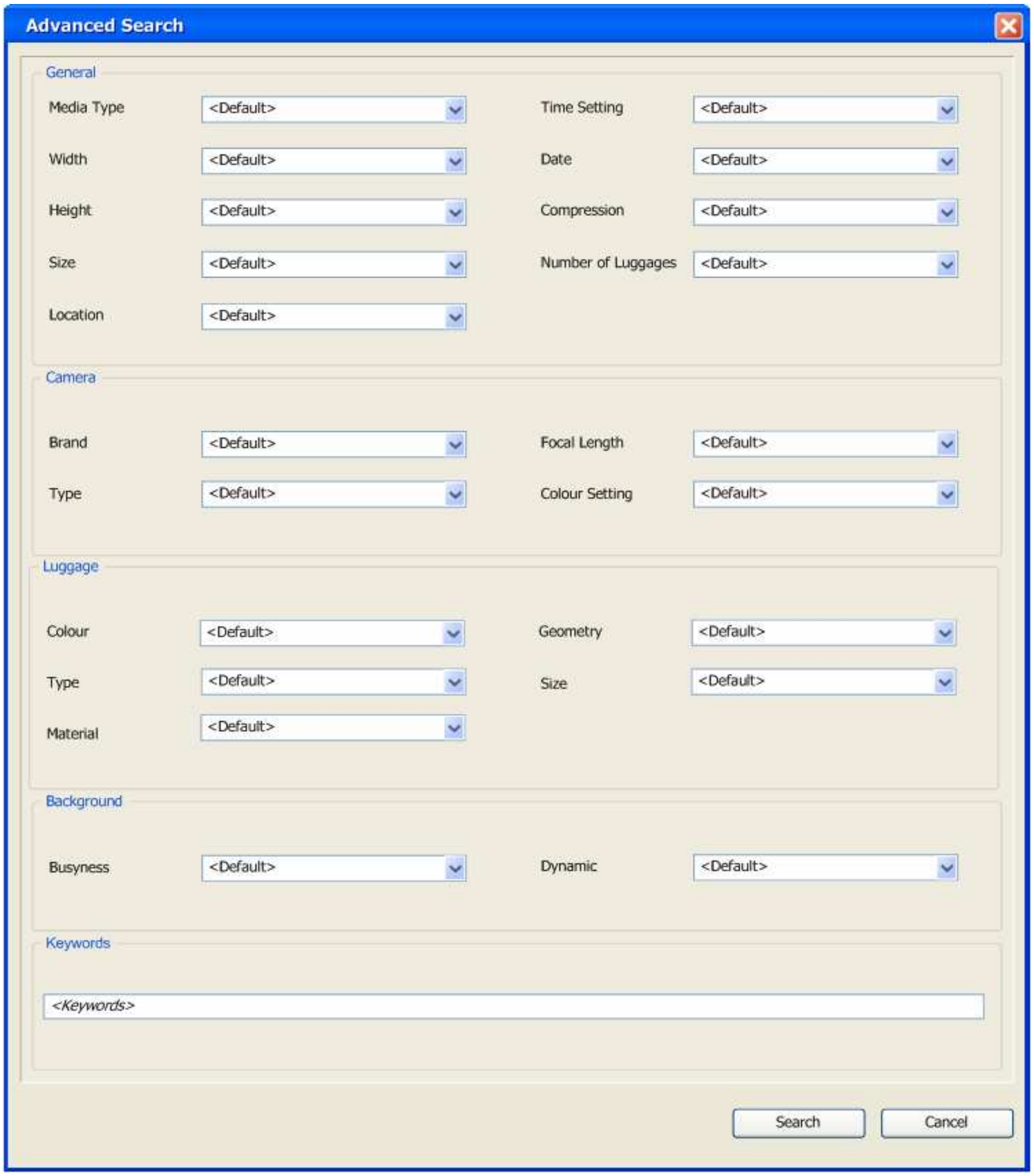


Remove

Are you sure you want to remove this media file from the database?

Figure C.4: Remove GUI

When a user wants to search for a particular picture or video, he can press the “Advanced Search” button. The screen in figure C.5 pops up and the user has more search options.



Advanced Search

General

Media Type <Default>
Width <Default>
Height <Default>
Size <Default>
Location <Default>
Time Setting <Default>
Date <Default>
Compression <Default>
Number of Luggages <Default>

Camera

Brand <Default>
Type <Default>
Focal Length <Default>
Colour Setting <Default>

Luggage

Colour <Default>
Type <Default>
Material <Default>
Geometry <Default>
Size <Default>

Background

Busyness <Default>
Dynamic <Default>

Keywords

<Keywords>

Figure C.5: Advanced search GUI

C.3 Connection with C#

UML database connection with C# is given in figure C.6.

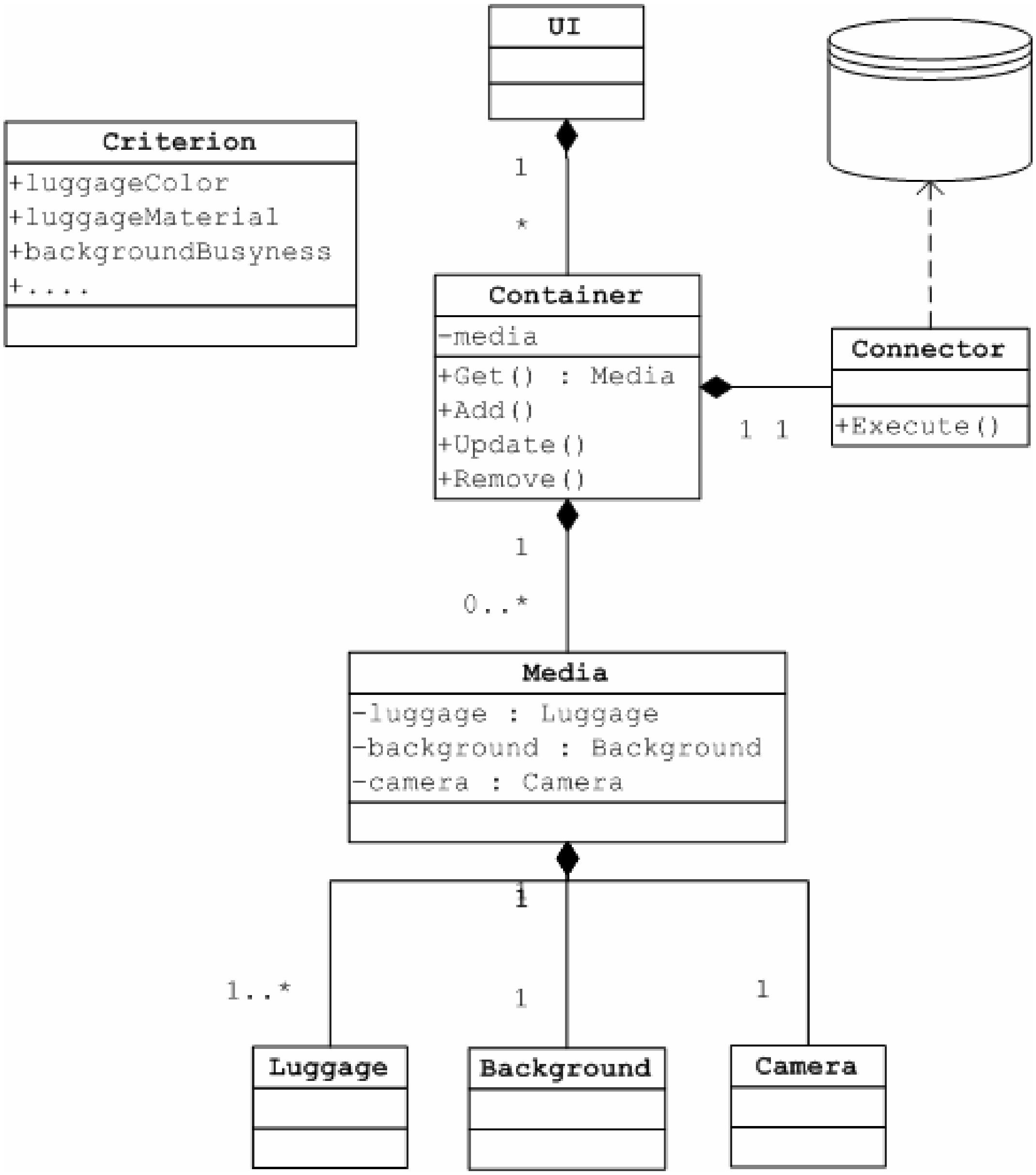


Figure C.6: Database connection with C#

Appendix

D

Movie Script

D.1 Introduction

To develop a system for recognizing and tracking luggage, all possible circumstances of the traveller-luggage interactions should be included. Therefore all possible situations have been written as movie scripts, which will be filmed later on.

Those scripts are divided into normal and suspicious behaviour. Firstly, a brief overview of the camera settings, background settings and walking routes will be given. Next, the scripts of the normal behaviour will be given, followed by the suspicious behaviour of the travellers accompanied by their luggage.

D.2 Camera Settings

The setting of the camera should be set once at the beginning of the shooting session. This is for simulating cameras that are hanging in trains and/or station platforms. The following settings should be set before the shooting session starts:

- height
- angle
- focal length
- area interest/Location

D.3 Background Settings

Depending on the script that will be shot and the elements that will be used, the following background settings may be used:

- dynamic background
    - walking travellers
    - moving objects, like baby wagon, pets etc.
- static background
- other objects in the scene (non-luggage)
    - static objects that are part of the surroundings, like seats, garbage etc.
    - dynamic objects, like pets, baby's etc.

D.4 Walking Routes

Walking routes describe the main routes where the actor will walk.

- Actor moving towards or away from the camera, Z axis
- Actor moving horizontal from camera, X axis
- Combination of actor moving in X and Z axis, rotation

D.5 Movie Scripts

In this section, all the possible circumstances of the traveller-luggage interactions are summarized as movie scripts. This is divided into normal and suspicious behaviour.

D.5.1 Normal Behaviour

A traveller, who is carrying a luggage, is having a normal behaviour when he/she shows no unusual action(s) with the luggage at the station/platform. In this section, several scripts containing the normal behaviour of a traveller along with his/her luggage will be described.

D.5.1.1 Initial

Script number: 1.1A
Scenario: initial empty
Location: station, platform
Elements: -
Start state: an empty station, platform
End state: an empty station, platform
Story: -

Script number: 1.1B
Scenario: initial crowded
Location: station, platform
Elements: -
Start state: a crowded station, platform
End state: a crowded station, platform
Story: -

D.5.1.2 Traveller Moves with Luggage

Script number: 1.2A
Scenario: a traveller who walks with his luggage
Location: station, platform
Elements: 1 actor & 1 luggage/ multiple luggage
Start state: the actor walks from one edge of the camera range
End state: the actor stops when he/she is a different edge of the camera range
Story:

- The actor along with his luggage starts to walk from one edge of the camera range
- The actor becomes fully visible to the camera range
- The actor walks towards a different edge of the camera range

- The actor becomes fully invisible to the camera

Script number: 1.2B
Scenario: a traveller who races with his luggage
Location: station, platform
Elements: 1 actor & 1 luggage/ multiple luggage
Start state: the actor races from one edge of the camera range
End state: the actor stops when he/she is at a different edge of the camera range
Story:

- The actor along with his luggage starts to race from one edge of camera range
- The actor becomes fully visible to the camera range
- The actor races towards a different edge of the camera range
- The actor becomes fully invisible to the camera

D.5.1.3 Traveller Places the Luggage Next to Him

Script number: 1.3A
Scenario: luggage next to owner
Location: station, platform
Elements: 1 actor & 1 luggage/ multiple luggage
Start state: the actor walks in/at the station/platform
End state: the actor stands next to its luggage
Story:

- The actor walks in/at station/platform along with his luggage
- The actor stops and places his luggage next to him
- The actor stands next to his luggage

Script number: 1.3B
Scenario: luggage behind the owner
Location: station, platform
Elements: 1 actor & 1 luggage/ multiple luggage
Start state: the actor walks in/at station/platform
End state: the actor stands in front of his luggage
Story:

- The actor walks in/at station/platform along with his luggage
- The actor stops and places his luggage behind him
- The actor stands in front of his luggage
- The luggage disappears from camera sight

D.5.1.4 Traveller Carrying Luggage Stands Still

Script number: 1.4A
Scenario: stationary owner with stationary luggage
Location: station, platform
Elements: 1 actor & 1 luggage
Start state: the actor stands still holding a luggage at his side
End state: -
Story:

- The actor starts fully visible to the camera holding his luggage at his side
- The actor stands still at a random place within camera view

Script number: 1.4B
Scenario: stationary owner with stationary luggage
Location: station, platform
Elements: 2 actors & 2 luggage / multiple actors & multiple luggage
Start state: the actors stand still holding a luggage at their side
End state: -
Story:

- The actors, holding their luggage at their side start fully visible to the camera
- The actors stand still at a random place within camera view

D.5.1.5 Traveller Walking In and Out the Scene

Script number: 1.5A
Scenario: traveller without luggage
Location: station, platform
Elements: 1/multiple actor
Start state: actor(s) walks in the scene
End state: -
Story:

- The actor(s) walk(s) in the scene for a period

Script number: 1.5B
Scenario: travellers without luggage, one traveller entering scene with luggage
Location: station, platform
Elements: multiple actors & 1 luggage
Start state: the actors with no luggage walk in scene
End state: a certain actor along with a luggage enters the scene
Story:

- Group of actors without luggage walk in the scene
- After a while, another actor with a luggage enters the scene
- The actor along with a luggage stays in the scene for a while
- The actor with a luggage leaves the scene

Script number: 1.5C
Scenario: traveller with a luggage goes into the camera range
Location: station, platform
Elements: 1 actor & 1 luggage
Start state: the actor carrying a luggage is walking into the camera range
End state: the actor with his luggage is fully visible to camera
Story:

- The actor who carries a luggage starts to walk from a certain place outside the camera view
- The actor reaches the edge of the range
- The actor along with his the luggage walks into the camera range

Script number: 1.5D
Scenario: travellers with a luggage are going into the range of the camera
Location: station, platform

Elements: 2 actors & 2 luggage / multiple actors & multiple luggage
Start state: actors carrying a luggage are walking into the range of the camera
End state: the actors and the luggage are fully visible to the camera
Story:

- The actors who carry a luggage start to walk from a certain place outside the camera view
- The actors reach the edge of the range
- The actors, along with their the luggage, walk into the range of the camera

Script number: 1.5E
Scenario: traveller with a luggage is going out of the range of the camera
Location: station, platform
Elements: 1 actor & 1 luggage
Start state: actor carrying a luggage is walking out of the range of the camera
End state: the actor with his luggage is fully invisible to camera
Story:

- The actor who carries a luggage starts to walk from a certain place inside the range
- The actor reaches the edge of the range
- The actor with the luggage at his side walks out of the range of the camera

Script number: 1.5F
Scenario: travellers with a luggage are going out of the range of the camera
Location: station, platform
Elements: 2 actors & 2 luggage / multiple actors & multiple luggage
Start state: actors carrying a luggage are walking out of the range of the camera
End state: the actors with their luggage is fully invisible to camera
Story:

- The actors who carry a luggage start to walk from a certain place inside the camera view
- The actors reach the edge of the range
- The actors, along with their the luggage, walk out of the range of the camera

D.5.1.6 Traveller Who Sits on the Bench

Script number: 1.6A
Scenario: traveller puts his luggage besides him on the bench
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene with the luggage
Story:

- The actor walks, along with his luggage, in to the nearest bench on the platform or station
- The actor sits down and puts his luggage besides him on the bench
- After 3 minutes, the actor grabs his luggage and gets to his feet
- The actor steps into the train or leaves the station or platform

Script number: 1.6B
Scenario: traveller puts his luggage besides him on the ground
Location: station, platform

Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene with the luggage
Story:

- The actor walks, along with his luggage in, to the nearest bench on the platform or station
- The actor sits down and puts his luggage besides him on the ground
- After 3 minutes, the actor grabs his luggage and gets to his feet
- The actor steps into the train or leaves the station or platform

Script number: 1.6C
Scenario: traveller puts his luggage on his laps
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene with the luggage
Story:

- The actor walks, along with his luggage, in to the nearest bench on the platform or station
- The actor sits down and puts his luggage on his laps
- After 3 minutes, the actor grabs his luggage and gets to his feet
- The actor steps into the train or leaves the station or platform

D.5.1.7 Luggage In the Luggage Space

Script number: 1.7A
Scenario: traveller puts his luggage in the luggage space of the train
Location: in the train
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene with the luggage
Story:

- The actor walks, along with his luggage, to the nearest available seat in the train
- The actor places his luggage in the luggage space above the seat
- The actor takes a seat
- After n minutes, the actor gets to his feet and grabs his luggage
- The actor leaves the scene along with the luggage

D.5.2 Suspicious Behaviour

In addition to the normal behaviour, travellers with a suspicious behaviour are those who handle their luggage very strangely. For such behaviour, the following scripts are made.

D.5.2.1 Abandoned Luggage

Script number: 2.1A
Scenario: abandoned luggage
Location: station, platform
Elements: 1 actor & 1 luggage
Start state: actor along with a luggage stands still

End state: actor drops the luggage and leaves the scene
Story:

- The actor, along with a luggage, starts fully visible to the camera
- The actor drops the luggage
- The actor walks out of the camera view

Script number: 2.1B
Scenario: abandoned luggage
Location: station, platform
Elements: 2 actors & 2 luggage / multiple actors & multiple luggage
Start state: the actors are holding the luggage
End state: the actors drop the luggage and leave the scene
Story:

- The actors, fully visible to the camera, walk in the scene
- Actors drop their luggage
- The actors walk out of the screen

Script number: 2.1C
Scenario: abandoned luggage
Location: station, platform
Elements: multiple actors & multiple luggage
Start state: multiple actors walk in the scene
End state: a particular actor drops a luggage and leaves the scene
Story:

- All actors walk in the scene
- One actor drops the luggage
- The actor walks out of the screen

Script number: 2.1D
Scenario: abandoned luggage
Location: station, platform
Elements: multiple actors & multiple luggage
Start state: 1 actor carrying n luggage
End state: actor drops all luggage or some luggage and leaves the scene
Story:

- All actors walk in the scene
- One actor drops the luggage
- The actor walks out of the screen

Script number: 2.1E
Scenario: abandoned luggage
Location: station, platform
Elements: 1 actor & 1 luggage
Start state: the actor stands still and carries a luggage
End state: the actor drops the luggage at a random traveller and leaves the scene
Story:

- The actor starts fully visible to the camera holding his luggage at his side
- The actor walks to a random fellow traveller

- The actor drops the luggage at his side
- The actor walks out of the screen

D.5.2.2 Throwing Luggage

Script number: 2.2A
Scenario: luggage throwing
Location: station, platform
Elements: 1 actor, 1/multiple luggage
Start state: the actor holds 1/multiple luggage
End state: the actor leaves the scene
Story:

- The actor enters the scene holding his/her luggage
- The actor throws one or more luggage away
- The actor walks out of the screen

D.5.2.3 Luggage Exchange

Script number: 2.3A
Scenario: luggage exchange
Location: station, platform
Elements: 2 actors & 1 luggage
Start state: one actor holds the luggage
End state: actors have exchanged the luggage and leave the scene
Story:

- Both actors walk into the scene from both side of the camera range
- Actor A gives the luggage to actor B
- Both actors walk out of the camera range

Script number: 2.3B
Scenario: luggage exchange
Location: station, platform
Elements: 2 actors & 2 luggage
Start state: both actors are holding the luggage
End state: the actors have exchanged the luggage and leave the scene
Story:

- Actor A and Actor B, each with a luggage, walk into the scene from both side of the camera range
- Actors walk towards each other and exchange the luggage when they cross
- The actors walk out of the screen

D.5.2.4 Abandoned Luggage on the Bench

Script number: 2.4A
Scenario: traveller leaves his luggage on the bench
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: The actor walks in the scene with the luggage
End state: The actor leaves the scene without the luggage
Story:

- The actor walks, along with his luggage, in to the nearest bench on the platform or station
- The actor sits down and puts his luggage besides him on the bench
- After 3 minutes gets to his feet
- The actor steps into the train or leaves the station or platform

Script number: 2.4B
Scenario: traveller leaves his luggage on the ground
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene without the luggage
Story:

- The actor walks, along with his luggage in, to the nearest bench on the platform or station
- The actor sits down and puts his luggage besides him on the ground
- After 3 minutes gets to his feet
- The actor steps into the train or leaves the station or platform

Script number: 2.4C
Scenario: traveller puts his luggage on his laps and leaves without it
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene without the luggage
Story:

- The actor walks, along with his luggage, in to the nearest bench on the platform or station
- The actor sits down and puts his luggage on his laps
- After 3 minutes, the actor drops the luggage on the bench and gets to his feet
- The actor steps into the train or leaves the station or platform

Script number: 2.4D
Scenario: traveller puts his luggage on his laps and leaves without it
Location: station, platform
Elements: 1 actor, 1 luggage
Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene without the luggage
Story:

- The actor walks, along with his luggage, in to the nearest bench on the platform or station
- The actor sits down and puts his luggage on his laps
- After 3 minutes, the actor drops the luggage on the ground and gets to his feet
- The actor steps into the train or leaves the station or platform

D.5.2.5 Abandoned Luggage in the Luggage Space

Script number: 2.5A
Scenario: traveller leaves his luggage in the luggage space of the train
Location: in the train
Elements: 1 actor, 1 luggage

Start state: the actor walks in the scene with the luggage
End state: the actor leaves the scene with the luggage
Story:

- The actor walks, along with his luggage, to the nearest available seat in the train
- The actor places his luggage in the luggage space above the seat
- The actor takes a seat
- After n minutes, the actor gets to his feet
- The actor leaves the scene along without the luggage

Appendix E Questionnaire

TU Delft: iBlobU Vragenlijst

Version 2.1

Doel:

In ons onderzoek naar agressiedetectie willen we een systeem ontwikkelen die het mogelijk maakt om achtergelaten bagage/pakketjes te detecteren. Omdat het onderzoek zich vooral richt op de omgeving van het station en in de trein wilden wij graag een kort interview houden met het personeel van de beveiliging. De informatie verkregen uit het interview zal gebruik worden om het systeem te ontwerpen. Ter voorbereiding van het afgesproken interview die gepland staat op 30 mei hebben wij hieronder een aantal vragen voorbereid.

Vragen:

- **Algemeen**
    1. Kunt u een korte beschrijving geven van uw taken?
    2. Hoeveel personen werken er in de controle kamer?
    3. Hoe lang zitten medewerkers van de beveiliging gemiddeld per dag achter de camera beelden?

- **Verdacht Gedrag**
    1. Waar let u op als u de camera beelden bekijkt?
    2. Hoe vaak komt het voor dat bagage achtergelaten wordt?
        1. In de trein?
        2. Op het station?
    3. Waar worden de bagages meestal achtergelaten?
    4. Is het moeilijk voor u om verdacht pakketjes/bagage to ontdekken?
    5. Wanneer wordt er besloten of een bagage/pakket verdacht is?
    6. Behalve dat bagage wordt achtergelaten, wat zijn de andere gevallen waar u op moet letten?
        1. Wat is de prioriteit?

- **Organisatie**
    1. Wat zijn de handelingen die u moet verrichten als er achtergelaten bagage wordt gevonden?
        1. Werkt u samen met andere organisaties?
        2. zijn er vaste regels/procedures?
        3. wat vindt u van deze procedures? omslachtig, goed, etc.?
    2. Wat is de awarenesstraining van de NS en hoe helpt de training?
        1. Hoe vaak wordt de training gegeven?
        2. Wat leert u van de training?

- **Techniek**
    1. Welke hulpmiddelen heeft u tot beschikking om camera beelden te analyseren?
    2. Wat kan volgens u beter in het huidige systeem?
    3. Bent u bekend met het systeem INES?
        1. Wat is INES?
        2. Wat vindt u van INES?
    4. Hoe kan een automatisch systeem uw huidige werk verlichten?
    5. Wat zijn de primaire eisen voor het automatische systeem?
    6. Wat is uw verwachting van het automatisch systeem, hoe kan dit systeem de huidige situatie verbeteren?

- **Juridische Zaken**
    1. Mag iedereen de videobeelden bekijken?
    2. Worden de videobeelden bewaard?
        1. Zo ja, Hoe lang worden de videobeelden bewaard?
    3. Komen er bepaalde procedures aan te pas om de privacy van de personen in de video te behouden?

Appendix

F

Interview NS Amsterdam

F.1 Inleiding

Op de TU-Delft loopt een onderzoek op het gebied van agressie detectie. In het kader hiervan is een project gestart naar achtergelaten bagage in en rondom de trein en het station. Een belangrijk deel van het onderzoek is het verkrijgen van het benodigde inzicht en expertise voor het detecteren van verdacht gedrag en achtergelaten bagage.

Met behulp van deze expertise is een beter ontwerp van een systeem mogelijk. Verder zal deze informatie waardevol zijn voor toekomstige onderzoeken op het gebied van verdacht gedrag en agressie detectie.

Om deze expertise te verkrijgen is er een interview gehouden met een aantal specialisten die voor de beveiliging zorgen op en rondom de trein stations van de NS. Wij willen graag mevrouw Monique Holland en de heer Dietz bedanken voor de bemiddeling van het interview. Uiteraard willen we ook de heer De Vries, Roffel en Van Teeffelen bedanken voor hun tijd om hun ervaring en expertise met ons te delen.

F.2 Procedure Interview

Datum: 30 mei 2007
Locatie: Amsterdam Centraal station
Tijd: 12:00 – 14:00

Ter oriëntatie van het interview is er van tevoren een gecategoriseerde lijst van vragen gemaakt. De drie geïnterviewden werden parallel door groepjes van twee personen geïnterviewd. De vragenlijst werd tijdens het interviewen slechts gebruikt als hulpmiddel voor het interview. De resultaten van de interviews zijn vervolgens samengebracht en uitgewerkt in dit verslag.

Geïnterviewde personen

Martin de Vries	: Projectleider
Mark van Teeffelen	: Hoofdbeveiliger
Cees Roffel	: Hoofdbeveiliger

F.3 Interview Resultaten

F.3.1 Algemeen

De beveiliging, Securitas, werkt voor de NS. Securitas werkt samen met de politie en de spoorwegpolitie.

1. Kunt u een korte beschrijving geven van uw taken?

De taak is het geven van leiding aan de operationele gang van zaken. De dag begint met briefing, waarin alle hoofdbeveiligers op de hoogte worden gesteld van problemen en bijzonderheden van de vorige en huidige dag. De taken worden verdeeld aan de verantwoordelijke personen. Problemen worden bij de desbetreffende persoon gemeld. Gehele dag door wordt alles bijgehouden in een log. Aan het einde van de dag wordt er een briefing gehouden waarin alle problemen die zich heeft afgespeeld besproken worden. Tenslotte wordt een keer per maand met de NS een vergadering gehouden over de stand van zaken.

2. Hoeveel personen werken er in de controle kamer?

Er werken 6 personen in de controle kamer op een normale dag. Voor elk shift geldt dat er te allen tijde drie personen in de controle kamer zitten, deze hebben de taak om de camerabeelden te analyseren. Twee personen surveilleren op het station en één hoofdbeveiliger neemt plaats achter het bureau. De taken van de mensen die achter het werkstation zitten, is meer dan alleen het analyseren van camerabeelden. Andere taken die door hen ook afgehandeld moeten worden zijn onder andere het beantwoorden van telefoons, verhelpen van storingen aan liften en verder vallen huishoudelijke taken ook onder hun rekening, zoals een opgesloten persoon op het toilet. Verder geldt dat er op feestdagen een hogere bezetting aanwezig is.

3. Hoe lang zitten medewerkers van de beveiliging gemiddeld per dag achter de camera beelden?

Een beveiliger zit gemiddeld per shift anderhalf uur en maximaal twee uur achter de camera beelden. In een dagelijkse routine draait een persoon normaal twee shiften achter de camera beelden, hierna gaan deze personen surveilleren op het station. Alleen in uitzonderlijke gevallen draaien mensen drie shifts. Per dag besteedt een beveiliger dus gemiddeld drie uur achter het werkstation. Na drie uur bestaat er een kans dat de concentratie van de beveiligers verminderd wordt.

F.3.2 Verdacht Gedrag

1. Waar let u op als u de camera beelden bekijkt?

Te allen tijde wordt de veiligheid van de reiziger gewaarborgd. Er wordt in het algemeen gelet op de volgende punten:

- afwijkend gedrag
- onbeheerde goederen
- grote groep mensen
- potentiële zakkenrollers

Door de surveillanten kunnen ook verdachte personen doorgegeven worden, deze worden dan ook in de gaten gehouden door de beveiligers achter de werkstations. Voorbeelden van afwijkend gedrag:

1. Te lang op het station wachten

   Normaal gaan de reizigers naar / uit de trein. Als iemand te lang op het station blijft, dan zal er waarschijnlijk iets aan de hand zijn. In een normale situatie blijft een reiziger gemiddeld 10 á 30 minuten op het station.

2. Mensen die opzettelijk tegen de stroom inlopen en hierbij de reizigers belemmeren

   De houding van de persoon wordt geobserveerd. Dit duidt mogelijk op een poging tot diefstal of andere crimineel gedrag.

3. Kijkrichting van een grote groep mensen

   Er is iets die hun aandacht trekt waar ze normaliter geen acht op slaan.

4. Verdachten die in een groep van twee personen werken

   Een persoon trekt de aandacht van de passagier in de trein door op het raam te kloppen, vervolgens pakt de andere verdachte de bagage van de afgeleide passagier.

5. Conducteur die uit de trein stapt met andere personen

   Die personen praten met de conducteur en er worden heel drukke gebaren gemaakt.

6. Een groep toeristen waarbij een verdachte persoon met een ander nationaliteit met de groep meeloopt

   Een verdachte persoon die meeloopt met deze groep toeristen is mogelijk bezig met het beroven van de toeristen.

7. Een passagier gaat te voorzichtig om met zijn bagage

   De inhoud van de bagage is waarschijnlijk waardevol of gevaarlijk.

2. Hoe vaak komt het voor dat bagage achtergelaten wordt?

In een uur ongeveer 1 of 2 keer.

3. Waar worden de bagages meestal achtergelaten?

Op het station wordt bagage vaak vergeten bij servicepointes, bij kaartverkooppunten en afdeling bagage.

4. Wanneer wordt er besloten of een bagage/pakket verdacht is?

Wanneer de reiziger niet op zijn bagage let wordt het al als verdacht beschouwd. Als dat werkelijk bewust/onbewust achtergelaten wordt, wordt de eigenaar gezocht door terug te kijken naar de videobeelden.

F.3.3 Organisatie

Draaiboeken voor verschillende situaties liggen te allen tijde klaar ter referentie, deze worden in praktijk niet vaak gebruik, dit komt doordat de mensen ervaring met de situaties hebben.

1. ***Wat zijn de handelingen die u moet verrichten als er achtergelaten bagage wordt gevonden?***

Wanneer er een melding gemaakt wordt door een surveillant, dan wordt het als het mogelijk is in beeld gebracht. Vervolgens wordt de bagage op gepaste afstand in de gaten gehouden voor 5 minuten. Als de eigenaar niet terugkomt, melden de beveiliging het aan de spoorwegpolitie en deze besluit of de bagage bij de afdeling gevonden voorwerpen ingeleverd kan worden of dat het verder onderzocht moet worden. Wanneer achtergelaten bagage gevonden wordt met behulp van de camera beelden, dan zal er vanuit de controlekamer surveillanten daarheen gestuurd worden, hierbij gelden dezelfde procedure als hierboven beschreven.

2. ***Werkt u samen met andere organisaties?***

Er wordt samengewerkt met:

- Spoorwegpolitie
- NS
- Connexxion
- Politie, waarbij veel contact wordt gemaakt met het CMK (Centrale Meld Kamer)
- Toezicht en Veiligheid

Met deze organisaties worden de briefings gehouden.

3. ***Wat vindt u van deze procedures? omslachtig, goed, etc.?***

Sommigen vinden de huidige procedure goed, maar andere vinden dat het soms te snel of te gemakkelijk gaat. Een terroristische aanslag komt vaak van binnenuit en er moeten duidelijkere afspraken komen over wat en wanneer een handeling gedaan moet worden.

4. ***Wat is de awareness-training van de NS en hoe helpt de training?***

De awareness-training is een workshop dat de NS heeft georganiseerd voor haar werknemers, om ze bewust te maken van mogelijke situaties op en rond het station en hoe ze hiermee om moeten gaan.

F.3.4 Techniek

1. ***Welke hulpmiddelen heeft u tot uw beschikking om camera beelden te analyseren?***

Het personeel in de controle kamer hebben de beschikking over drie systemen:

- ATM

  Het ATM systeem is het oudste systeem en meest gebruikte systeem, dit is het systeem waar de beveiligers dagelijks achter zitten. Men beschikt per persoon over 8 schermen, 6 van deze schermen zijn camera beelden, die samen tot 1 cameragroep behoren van het betreffende station. De overige twee schermen worden gebruikt voor het selecteren van stations en camera's, het bekijken van plattegronden, bekijken van statistieken en het bijhouden van het logboek. De beelden van de ATM zijn analoge beelden, dit geeft de mogelijkheid om in te zoomen met de DOME camera tot een zeker niveau. Indien nodig kan het systeem ook teruggaan in de tijd en de opgeslagen beelden weergeven. Behalve DOME camera's zijn er per station ook 16 risicocamera's, beelden van deze camera's worden 24 uur opgeslagen. Hoewel de camerabeelden van de risicocamera's worden opgeslagen, is er nog steeds het probleem dat tijdens de camerashifts de beelden van deze 16 camera's niet parallel worden opgenomen, maar achterelkaar worden geplakt op 1 tape. Dit heeft als resultaat dat er een zwart gat ontstaat bij elke shift, na de shift heeft men namelijk geen zicht meer op de beelden van de camera's voor de shift, en belangrijker deze beelden worden verder niet meer op de achtergrond opgenomen. Dit heeft als resultaat dat hoewel camera's gemarkeerd worden als risicocamera's er niet 24 uur van alle camera's opnames gemaakt worden. Tenslotte kan het systeem ook beelden doorsturen naar andere controlekamers elders in het land.

- ISDN

  Het ISDN systeem wordt in principe niet gebruikt, dit heeft te maken met de lage bandbreedte die het systeem tot zijn beschikking heeft. Om camerabeelden te bekijken moet de gebruiker eerst inloggen waarna een verbinding wordt gemaakt met de gekozen cameragroep zoals bij het ATM systeem. Door de lage verbindingsnelheid is ook meteen duidelijk waarom dit systeem niet bruikbaar is in vergelijking met het ATM systeem, de framerate van de camerabeelden is zeer laag en de kwaliteit van de beelden is ook zeer slecht in vergelijking met de beelden van het ATM systeem. De reden waarom het systeem toch gebruikt wordt is omdat het systeem beschikt over camera's die bij ATM in het systeem niet beschikbaar zijn.

- ISVL

  Het ISVL systeem is het nieuwste systeem dat nu door de NS in gebruik wordt genomen, het systeem is nog in ontwikkeling. Het ISVL verschilt van de andere twee systemen in het gebruik van de datacommunicatie, waar de andere twee systemen gebruik maken van een analoge verbinding, wordt hier gebruik gemaakt van IP waardoor alles digitaal wordt opgestuurd. Dit is het systeem dat ook in Utrecht gebruikt wordt. Een nadeel van het systeem is dat hoewel inzoomen mogelijk is, de kwaliteit hierdoor drastisch vermindert. Dit is het resultaat van het gebruik van digitale camerabeelden.

Het analyseren van de camerabeelden wordt door het personeel gedaan, dit zorgt ervoor dat het analyseren subjectief is. Ook is de ervaring van het personeel van belang. Wanneer er een beveiliger twijfelt, zal die aan een collega om zijn mening vragen.

2. *Wat kan volgens u beter in het huidige systeem?*

Een snellere verbinding van internet, camera met geluid, alert systeem (het detecteren van afwijkend gedrag), van alle camera's zouden opnames gemaakt moeten worden en hogere framerate.

3. *Bent u bekend met het systeem INES?*

Sommige zijn wel op de hoogte van INES systeem, maar hebben geen ervaring ermee. Andere hebben enig ervaring met het systeem.

4. *Wat is INES?*

INES is een project dat gebruik maakt van smart camera's die ook geluid hebben die reageren op verdachte situaties. De smart camera's zijn getraind op het herkennen van zulke situaties, waarbij een verdachte situatie wordt herkend door dat de smart camera's weten hoe een normale situatie zou moeten zijn. Wanneer iemand voor een langere periode stilstaat omdat hij in slaap gevallen is geeft de smart camera een melding door alle andere beelden opzij te duwen en aan te geven dat de camera iets gedetecteerd heeft. De persoon die de camera beelden bekijkt kan dan zelf beslissen of het echt een verdachte situatie is of dat het een false alarm is. Het voordeel van de INES is dat er 24 uur per dag opgelet wordt, omdat de camera's pas geactiveerd worden wanneer er vreemd gedrag gedetecteerd wordt of abnormaal geschreeuw, zoals om hulp of wanneer er dode objecten gedetecteerd worden, dit kan een bagage zijn of een persoon dat een hartaanval heeft gehad of iemand die ligt te slapen.

5. *Wat vindt u van INES?*

Slechts een van de geïnterviewde heeft de enige ervaring met het INES project. Hij vindt het een goed systeem, ook is de software snel onder de knie te krijgen. INES zou wel verbeterd kunnen worden door ook gebruik te maken van luidsprekers, hiermee kan een beveiliger in de controle kamer een reiziger meteen waarschuwen. Een ander geïnterviewde heeft zelf niet meegedaan met het project, maar denkt dat het project te groot geprobeerd is en niet goed uit de verf is gekomen.

6. *Hoe kan een automatisch systeem uw huidige werk verlichten?*

Door verdacht gedrag te detecteren kan een automatisch systeem het huidige werk verlichten, ook kan een automatisch systeem gebruikt worden om te detecteren of de surveillant het station al geopend heeft en waarschuwen als dit niet zo is.

7. *Wat zijn de primaire eisen voor het automatische systeem?*

Het detecteren van verdachte situatie is het belangrijkst.

8. *Wat is uw verwachting van het automatische systeem, hoe kan dit systeem de huidige situatie verbeteren?*

Een van de geïnterviewden vindt dat wanneer INES goedkoper zou zijn, de NS zou aanschaffen en gebruiken. De andere geïnterviewden vinden dat er geen behoefte is aan gezichtsherkenning, maar wel het detecteren van onbeheerde goederen en verdacht gedrag. Het is belangrijk om veiligheid te waarborgen die een onderdeel is van sociale veiligheid. Sociale veiligheid is hoe de burger zich voelt en dit kan door middel van bijvoorbeeld het voorzien van voldoende verlichting, preventief handelen, goed zichtbare camera's, het regelmatig legen van prullenbakken en het voorkomen van graffiti.

F.3.5 Juridisch

1. *Mag iedereen de videobeelden bekijken?*

Nee, alleen geautoriseerde mogen de videobeelden bekijken. Personen die geautoriseerd worden, moeten eenmalig drie keer gescreend worden voordat hij/zij toegang tot de beelden krijgt.

2. *Worden de videobeelden bewaard?*

Ja, en wel gedurende zeven dagen. Daarna worden ze overschreven.

3. ***Wie bewaart de videobeelden?***

De videobeelden zijn eigendom van de NS en worden in de server bewaard. Elk station heeft zijn eigen server. De videobeelden worden door de beveiliging op CD's gezet en vervolgens naar de locatie van de server vervoerd. Als een opsporingsambtenaar de beelden wilt bekijken, dan dient hij toestemming te vragen aan de justitie.

4. ***Komen er bepaalde procedures aan te pas om de privacy van de personen in de video te behouden?***

NS neemt alleen de beelden op. Alleen met toestemming van de justitie mogen derde partijen de videobeelden bekijken.

F.4 Conclusie

Uit de resultaten van het interview komt duidelijk naar voren dat behalve het analyseren van camera beelden de mensen van de beveiliging met meer taken worden belast. Verder geldt ook dat het herkennen van verdacht gedrag sterk afhankelijk is van de expertise en ervaring van de persoon die de beelden analyseert, het is dan ook een vooral subjectief gebeuren. Een systeem dat automatisch verdacht gedrag kan herkennen zal de werklast van de beveiliging kunnen verminderen en personen met minder ervaring en expertise ondersteunen bij het detecteren van verdacht gedrag. Hiermee zal men sneller bekende scenario's van verdacht gedrag kunnen detecteren en vervolgens bijbehorende actie ondernemen.

Uitgevoerd in opdracht van:

TU Delft

Door:

iBlobU

Calvin Choy
May Chung
Gherry Harahap
Meriane Natadarma
Sin Lin Wu

Appendix G

Interview NS Utrecht

G.1 Introductie

Op de TU-Delft loopt een onderzoek op het gebied van agressie detectie. In het kader hiervan is een project gestart naar achtergelaten bagage in en rondom de trein en het station. Om een beter beeld te krijgen over de huidige situatie met betrekking tot het gebruik van systemen om de veiligheid te verbeteren, hebben wij een bezoek gebracht aan het hoofdkantoor van de NS te Utrecht. Tijdens het bezoek werden verschillende aspecten behandeld die als doel hebben om de sociale veiligheid in en rondom de trein en station te verbeteren. Deze aspecten zullen in dit verslag uitgewerkt worden en zal gebruikt kunnen worden bij toekomstig ontwikkeling van systemen op het gebied van agressie en veiligheid.

G.2 Acknowledgement

Wij willen graag de heer Heida en de heer Prins bedanken voor de rondleiding en hun tijd om hun expertise met ons te delen.

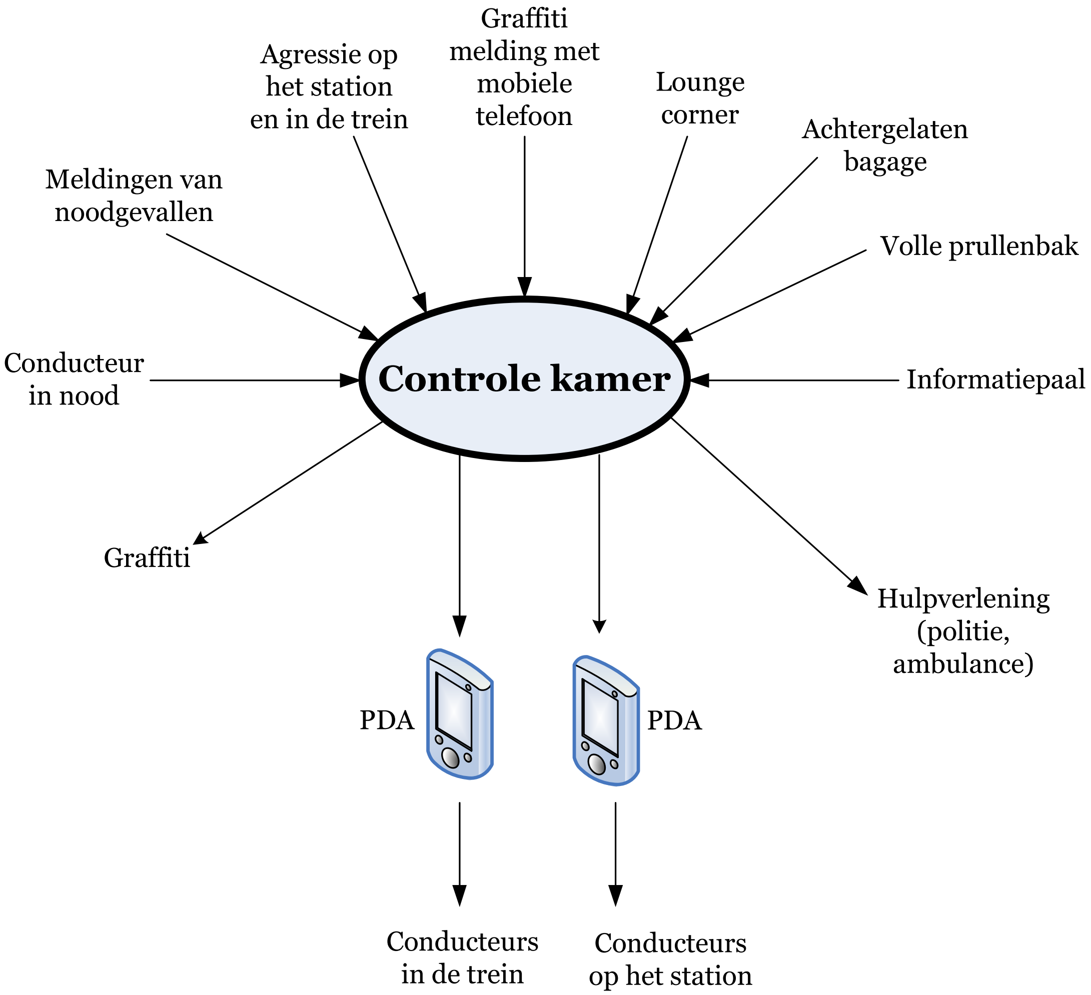
G.3 Interview Resultaten

G.3.1 Sociale Veiligheid

De veiligheid van de reizigers staat voorop bij de NS. Het is dus belangrijk dat de reizigers zich veilig voelen op het station en in de trein. Om deze veiligheid te waarborgen en te verbeteren is de NS begonnen met het INES project. Het INES staat voor Innovatieve Experimenten Sociale veiligheid. Met behulp van innovatieve technieken wil de NS verschillende aspecten aanpakken om de sociale veiligheid van de reizigers te verbeteren. Met het gebruik van innovatieve technieken wil de NS zich richten op de verhoging van de effectiviteit die behaald kan worden met behulp van nieuwe technieken en niet het meer inzetten van wat al aanwezig is (i.e. meer camera's en meer beveiligers). Het verbeteren van de sociale veiligheid kan op verschillende manieren gebeuren, dit heeft te maken met verschillende aspecten die invloed hebben op het geven van een veilig gevoel op het station. Zo kan naast het aanpakken van criminaliteit en agressie ook de sociale veiligheid verbeterd worden door de omgeving op het station schoon te houden. Uit onderzoek is gebleken dat een schone omgeving een veiliger gevoel geeft. Om de effectiviteit van het experiment te beoordelen moet de sociale veiligheid gemeten worden. Dit kan op twee manieren, doormiddel van statistieken, waarbij de objectiviteit belangrijk is en doormiddel van enquêtes waar het gevoel van veiligheid gemeten wordt bij de reizigers, door ook de subjectiviteit van de laatste manier te gebruiken krijgen ze de werkelijke meting. Om de resultaten te kunnen vergelijken van voor en na het experiment werden twee metingen gedaan, een nul meting voor het experiment en een nieuwe meting aan het eind van het experiment. Bij het vergelijken van de resultaten werden significante verschillen gedetecteerd.

G.3.2 Het INES Project

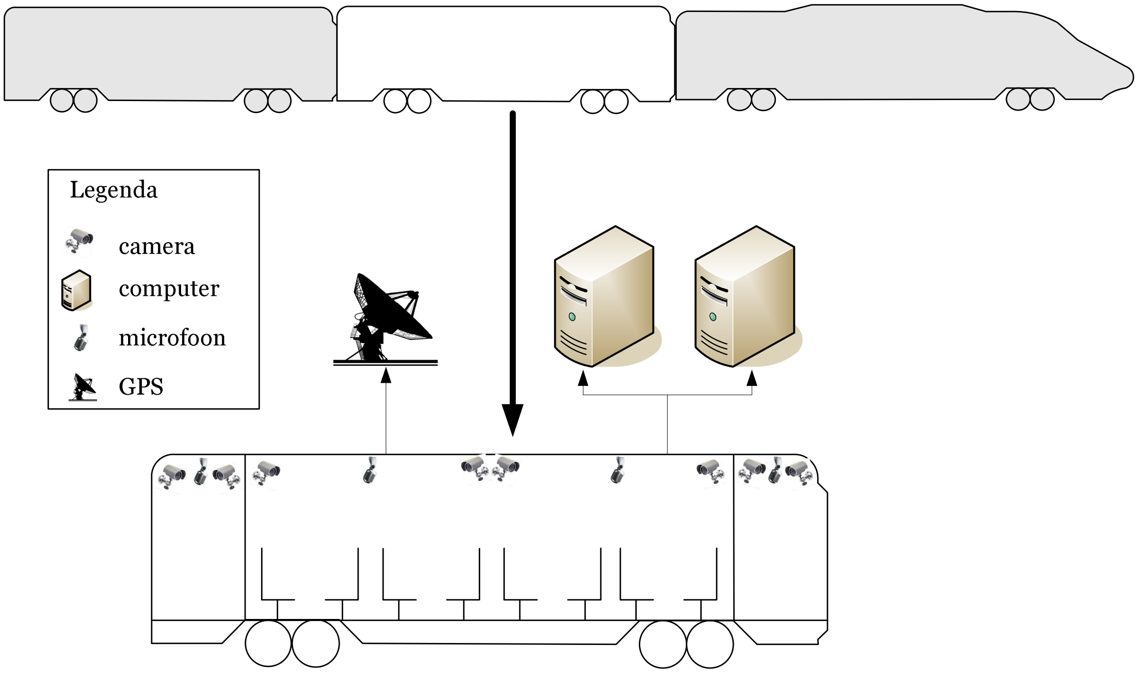
De toepassing van de verschillende technieken die gebruikt zijn tijdens het experiment om de sociale veiligheid, in en rondom de trein en het station te verbeteren, kan kort worden samengevat in het figuur hieronder (zie figuur G.1).


Graffiti melding met mobiele telefoon
Agressie op het station en in de trein
Lounge corner
Achtergelaten bagage
Meldingen van noodgevallen
Volle prullenbak
Conducteur in nood
Controle kamer
Informatiepaal
Graffiti
Hulpverlening (politie, ambulance)
PDA
PDA
Conducteurs in de trein
Conducteurs op het station

Figuur G. 1: Schema van controle kamer meldingen

G.3.2.1 Veiligheid in de trein

Om de veiligheid in treinen te verbeteren zijn er gedurende een half jaar elf internationale treinen gebruikt om een experiment uit te voeren. Deze treinen hebben een compartiment dat beschikt over GPS, acht camera's, vier microfoons en twee computers om video en geluid te verwerken. Elk microfoon is gekoppeld aan twee camera's. Omdat compartimenten tussen treinen worden verwisseld functioneren deze als een self supporting unit met elk een uniek id (zie figuur G.2).



Figuur G.2: Opstelling van de camera's en microfoons

Een van de belangrijkste punten die verbeterd moeten worden in de treinen is het voorkomen van agressie en de mogelijkheid geven om snel en veilig hulp te vragen indien nodig. Om dit te realiseren wordt er in de trein gebruik gemaakt van geluid analyse technieken om agressie te detecteren. Door eigenschappen van geluid te analyseren zoals frequentie en amplitude kan men agressie detecteren. Waneer het systeem agressie heeft gedetecteerd kan er een alarm gestuurd worden naar de centrale zodat men op tijd op de hoogte is van de situatie is en gepaste maatregelen kan nemen. Doordat de microfoons gekoppeld zijn aan twee camera's, kunnen de juiste beelden ook direct worden opgevraagd. Deze beelden kunnen eventueel gestuurd worden naar de PDA's van de conducteurs in de trein zodat deze ook op de hoogte worden gesteld van de situatie en meer voorbereid kunnen handelen.

Mocht de conducteur in de trein in discussie komen met een agressieve reiziger, dan is het uitgangspunt altijd om verdere escalatie van agressie te voorkomen. Hoewel dit in de meeste gevallen lukt zijn er altijd gevallen waarbij de conducteur zodanig bedreigd wordt dat hij of zij hulp nodig heeft. Ook in scenario's waarbij iemand in de trein onwel wordt en hulp nodig heeft is het belangrijk dat er zo snel mogelijk hulp gevraagd kan worden. Om dit probleem op te lossen heeft de NS op verschillende locaties in de treinen alarm knoppen geïnstalleerd. Hiermee krijg men direct contact met de centrale en kan er doormiddel van een microfoon gecommuniceerd worden. Verder beschikken de conducteurs over een apparaat waarmee hij doormiddel van een druk op de knop (onopgemerkt) om hulp kan vragen indien hij of zij wordt bedreigd en hulp nodig heeft.

G.3.2.2 Veiligheid op het Station

Om de veiligheid van de reiziger op en rondom het station te verbeteren, heeft de NS experimenten uitgevoerd om crimineel gedrag te voorkomen en reizigers geïnformeerd te houden over actuele zaken.

Figuur G.3: Graffiti in een station

Een van de ergernissen op en rondom het station is graffiti (zie figuur G.3). Om dit probleem op te lossen heeft de NS geëxperimenteerd met systemen die door middel van video content analyse kunnen detecteren of er graffiti wordt gemaakt. Doordat de omgeving van het systeem zich afspeelt rondom het station waar veel beweging aanwezig is geeft het systeem een hoog aantal false positives. Een ander manier om het graffiti probleem op te lossen werd landelijk getest. Doormiddel van mobiele telefoons konden mensen een bericht met foto sturen naar de centrale die vervolgens beveiligers kon sturen of doormiddel van een microfoon een waarschuwing kon geven.

Een ander belangrijk aspect is het op tijd detecteren van achtergelaten bagage. Doormiddel van technieken om beelden te analyseren kan men achtergelaten bagage detecteren zodat men in de controle kamer zo snel mogelijk de eigenaar kan traceren. Dit is vooral belangrijk bij verdachte pakketjes. Een andere toepassing van het analyseren van beelden is bijvoorbeeld het detecteren van overvolle vuilnisbakken.

Het is ook van belang dat reizigers altijd geïnformeerd zijn over actuele zaken die zich afspelen op en rondom het station en dat het station de reizigers een schoon en rustig gevoel geeft wat resulteert in een veiliger gevoel voor de omgeving. Om dit te bereiken heeft de NS op verschillende plekken op het station zuilen geplaatst waar actuele informatie wordt getoond zoals reistijden en het actuele nieuws. Verder worden er op sommige stations lounge corners geplaatst die een rustige en veilige uitstraling geven op het station en hierdoor criminele gedrag vermindert (zie figuur G.4). Ook het gebruik van schermen op het station om camera beelden te tonen werkt positief, men voelt zich veiliger en criminelen zijn er dan bewust dat ze in de gaten worden gehouden.

Figuur G.4: Lounge corner op het station

In geval van calamiteiten staan er overal op het station informatiepalen. Behalve dat men hier algemene informatie kan vragen, kan men in nood gevallen ook direct contact zoeken met de centrale en om hulp vragen. Verder zijn er op sommige stations toegangspoorten geïnstalleerd, hiermee kunnen alleen reizigers met een geldig vervoerbewijs doorheen. Het is mogelijk dat er in bepaalde situaties iedereen toegang tot de poortjes moet krijgen, denk hierbij aan ramp situaties of een groep hooligans die men zo snel uit het station wil hebben. Bij het ontwerp van het systeem is daardoor ook rekening gehouden dat deze poortjes allemaal van afstand te besturen zijn vanuit de centrale.

G.3.4 Privacyregels van de Controle Kamer in Utrecht

Door het gebruikt van veel beeld materiaal is het belangrijk dat de privacy van de reizigers gewaarborgd wordt. De beelden, die worden gemaakt door de camera's, worden daardoor slechts gedurende een week bewaard. Verder zijn deze beelden slechts toegankelijk voor bevoegden. Ook de toegang tot de centrale kamer waar de beelden worden bekeken is beperkt tot de mensen die daar in werken. Te allen tijde worden alle personen geregistreerd die de camera beelden bekijken.

G.4 Conclusie

Uit de meting die gemaakt is aan het eind van het experiment is gebleken dat het INES project heeft gezorgd voor een verbeterde sociale veiligheid bij de reizigers en dat criminaliteit is gedaald. Hoewel met de huidige technieken die gebruikt worden niet alle gewenste problemen opgelost kunnen worden, heeft het INES project toch zijn effectiviteit kunnen tonen in het experiment om de sociale veiligheid te verbeteren.

Uitgevoerd in opdracht van:

TUDelft

Door:

iBlobU

Calvin Choy
May Chung
Gherry Harahap
Meriane Natadarma
Sin Lin Wu