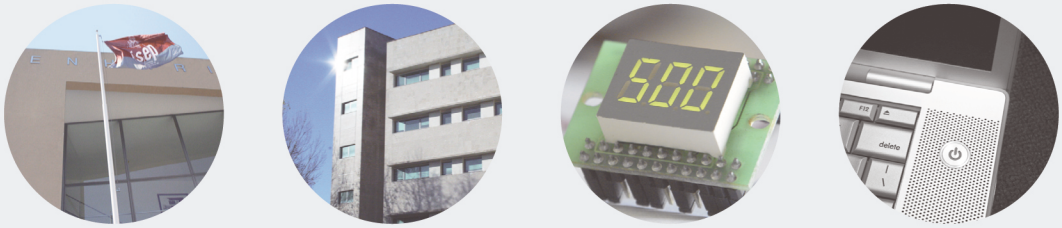
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



People Counting System using Existing Surveillance Video Camera

DIOGO MOREIRA CABRAL MACHADO
Novembro de 2011

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

DEE

People Counting System using Existing Surveillance Video Camera

Diogo M. Cabral Machado

1020936

— *Supervisor* —

Eduardo Alexandre Pereira Silva (PhD.)

— *Co-Supervisor* —

José Miguel Soares Almeida (MsC.)

ISEP/LSA

Thesis submitted under the

MsC. in Electrical and Computer Engineering

Autonomous System Profile

NOVEMBER, 2011

MsC. in Electrical and Computer Engineering

People Counting System Using Existing Surveillance Video Camera

by

Diogo M. Cabral Machado

Autonomous Systems Laboratory

Instituto Superior de Engenharia do Porto

MsC. thesis supervised by:

Eduardo Alexandre Pereira Silva, *PhD.*

Co-supervised by:

José Miguel Soares Almeida, *MsC.*

Autonomous System Laboratory

Instituto Superior de Engenharia do Porto

Porto, November 2011

Abstract

The Casa da Música Foundation, responsible for the management of Casa da Música do Porto building, has the need to obtain statistical data related to the number of building's visitors. This information is a valuable tool for the elaboration of periodical reports concerning the success of this cultural institution. For this reason it was necessary to develop a system capable of returning the number of visitors for a requested period of time.

This represents a complex task due to the building's unique architectural design, characterized by very large doors and halls, and the sudden large number of people that pass through them in moments preceding and proceeding the different activities occurring in the building.

To achieve the technical solution for this challenge, several image processing methods, for people detection with still cameras, were first studied. The next step was the development of a real time algorithm, using OpenCV libraries and computer vision concepts,to count individuals with the desired accuracy. This algorithm includes the scientific and technical knowledge acquired in the study of the previous methods. The themes developed in this thesis comprise the fields of background maintenance, shadow and highlight detection, and blob detection and tracking.

A graphical interface was also built, to help on the development, test and tunning of the proposed system, as a complement to the work.

Furthermore, tests to the system were also performed, to certify the proposed techniques against a set of limited circumstances. The results obtained revealed that the algorithm was successfully applied to count the number of people in complex environments with reliable accuracy.

Keywords

OpenCV, People Counting, Computer Vision, Background Maintenance, Segmentation.

This page was intentionally left blank.

Resumo

A Fundação Casa da Música, responsável pela gestão do edifício da Casa da Música, tem a necessidade de obter dados estatísticos relativos ao número de visitantes. Esta informação é uma ferramenta valiosa para a elaboração periódica de relatórios de afluência para a avaliação do sucesso desta instituição cultural. Por este motivo existe a necessidade da elaboração de um sistema capaz de fornecer o número de visitantes para um determinado período de tempo.

Esta tarefa é dificultada pelas características arquitetônicas, únicas do edifício, com portas largas e amplos *halls*, e devido ao súbito número de pessoas que passam por estas áreas em momentos que antecedem e procedem concertos, ou qualquer outras actividades.

Para alcançar uma solução técnica para este desafio foi inicialmente elaborado um estado da arte relativo a métodos de processamento de imagem para deteção de pessoas com câmeras de vídeo. O passo seguinte foi, utilizando bibliotecas de OpenCV e conceitos de visão computacional, o desenvolvimento de um algoritmo em tempo real para contar pessoas com a precisão desejada. Este algoritmo inclui o conhecimento científico e técnico adquirido em métodos previamente estudados. Os temas desenvolvidos nesta tese compreendem os campos de manutenção do fundo, deteção de zonas sub e sobre iluminadas e deteção e seguimento de *blobs*.

Foi também construida uma interface gráfica para ajudar o desenvolvimento, teste e afinação do sistema proposto como complemento ao trabalho desenvolvido.

Além disso, perante um conjunto limitado de circunstâncias, foram efectuados testes ao sistema em ordem a certificar as técnicas propostas. Os resultados obtidos revelaram que o algoritmo foi aplicado com sucesso para contar pessoas em ambientes complexos com precisão.

Palavras-Chave

OpenCV, Contagem de Pessoas, Visão Computacional, Manutenção de Fundo, Segmentação.

This page was intentionally left blank.

Acknowledgments

Many people contributed to this dissertation and I am gratefull to all of them.

I would like to thank my supervisors Eduardo Alexandre Pereira Silva and José Miguel Soares Almeida for their scientific guidance and valuable contributionsduring the couse of this work.

I thank Carlos Almeida and Guilherme Silva for their helpfull comments, knowledge, constructive discussions and support as friends.

Lastly, but most importantly, I offer my regards to my family that has made possible the completion of this project.

Diogo Cabral Machado

This page was intentionally left blank.

Contents

List of Figures

Acronyms

2D	*Bi-dimensional space*
3D	*Three-dimensional space*
BEM	*Background Estimation Module*
CRT	*Cathode Ray Tube*
EM	*Expectation Maximisation*
GIMP	*GNU Image Manipulation Program*
GNU	*GNU's Not Unix*
GTK	*GIMP Toolkit*
GUI	*Graphical User Interface*
IPP	*Integrated Performance Primitives*
ISEP	*Instituto Superior de Engenharia do Porto*
LED	*Light Emitting Diode*
LSA	*Laboratório de Sistemas Autónomos*
LTCBM	*Long Term Color-based Background Mode*
OpenCV	*Open Computer Vison*
R&D	*Research and Development*
RGB	*Red, Green, Blue*
SM	*Segmentation Module*
SMS	*Short Message Service*
SQL	*Structured English Query Language*
STCBM	*Short Term Color-based Background Model*
TCM	*Tracking and Counting Module*
XML	*Extensible Markup Language*

This page was intentionally left blank.

Chapter 1

Introduction

Contents

1.1 Motivation

The tracking of people that enter or pass trough a determined space is undoubtedly, an important tool for statistical and marketing research purposes, likewise for an increased security control (eg. emergency evacuation). In this work, the objective was to build a system to count the thousands of people that daily visit the Building of Casa da Música in Oporto City, with the purpose of obtaining statistical data.

The building Casa da Música do Porto is managed by Casa da Música Foundation-that is funded by both private and governmental capital. Like any other organization, it's management is required to make periodical reports, which success is strongly attached to the number of visitors that the building receives. This leeds to the need developing a people counting system to easily and reliably, assist valuable data for the elaboration of these reports.

In these days, the solutions presented in the market do not satisfy the needing requirements.

The Autonomous Systems Laboratory (LSA) R&D unit from ISEP, the Engineering School of Porto Polytechnic, conducts research projects in the field of distributed perception systems in complex environments.

In this project, LSA technical and scientific knowledge will be used to fulfill the needs of Casa da Música Foundation.

1.2 Addressed Problem

1.2.1 People Counting Challenge

In the past few years, the use of video cameras to track and count people increased considerably due to the advance in image processing algorithms and computer's technology.

Several systems and technologies exist to do such work, and because of it's importance for multitude of public spaces applications, there are many solutions offered by commercial products. The differences between these products can be seen as a pyramid of functionalities (Figure 1.2).

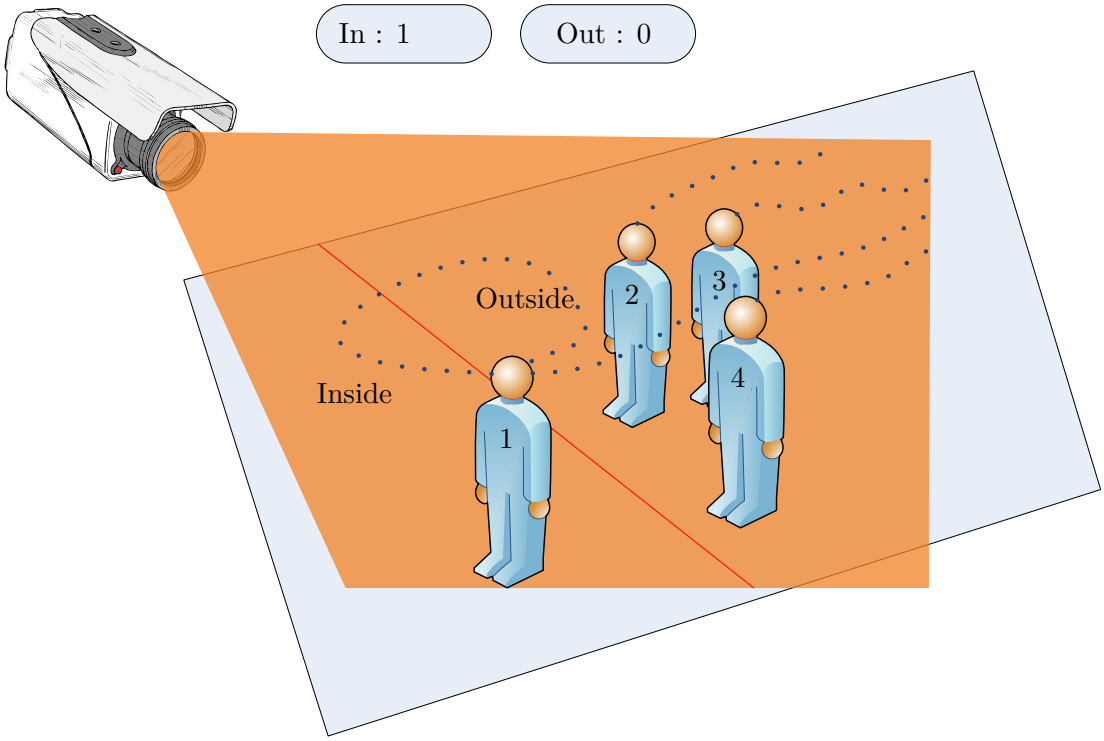


Figure 1.1: System Overview

- *Systems that count the number of people* are the most basic systems in terms of functionality. These systems can be an infra-red Light Emitting Diode (LED) and a receiver that counts one person when light from the LED to the receiver is interrupted, or it can be a laser beam based on the same principle. In both systems, the detection area should be a narrow door since these sensors are not able to distinguish when two people cross the area at the same time. Another type of sensor is the weight sensor, require heavy environmental modifications and significant maintenance.

- *Systems that determine people direction* can use the sensors above mentioned. In the case of the **LED** and the **laser beam** sensors, this is accomplished using a pair of sensors and checking which one is first interrupted. It's also possible to estimate people direction using **weight sensors**. In this group it also has the "de facto" system in accuracy - the **turnstile**. The drawback of this system is that it limits the traffic flow and creates the feeling of entrapment. There are also security concerns in using these systems as they create barriers in case of an emergency evacuation. Another solution are the **video cameras**, that can be used to count people entering or leaving through a door. This system is usually set

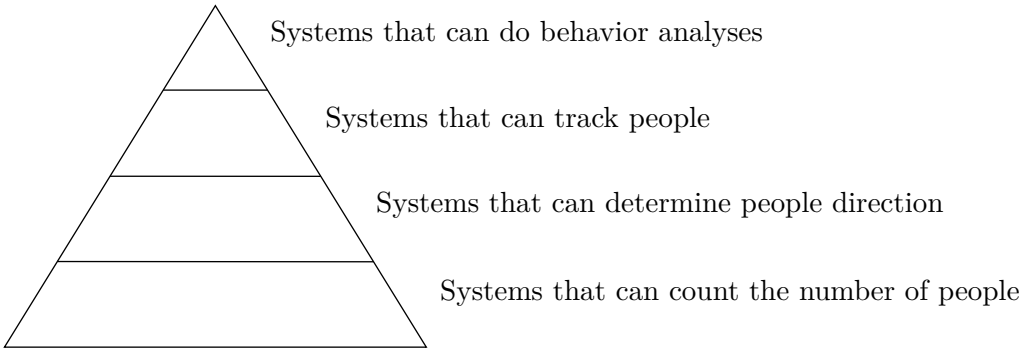


Figure 1.2: Different types of sensors

up in a wall, facing downwards to reduce the occlusions of people. Another type of sensor is the **thermal camera**, which image is filled with colors that illustrate the temperature of the objects in it's field of view. This sensor, however, becomes unusable when the environment temperature is close to that of the human body, since it becomes impossible to distinguish them.

- *Systems that track people* generally use video cameras for sensors. These systems can track individuals while they appear in the camera's field of view, resulting in a more accurate model of their actions. For instance, if a person is standing in the counting area (perhaps talking with a fellow coworker) it would be needed to track the movements of said person to accurately estimate the first and final positions beside the counting line/area. Some of this systems can even track people through the use of multiple cameras, with uses like following a person in a shopping mall using the surveillance cameras.

- Systems that do behavior analysis can use the tracking that was described in the previous item, to estimate different metrics about the observed positions over time and space. It can be used to monitor a building's emergency evacuation or to monitor in a shop, if a new product is having a lot of attention from the clients. The applications of this technology are much vaster than the simple people counter.

The counting of building visitors has a critical area, the main entrance, where it is very difficult with the existing technology to accurately, count the visitors with minimum visual and architectural impact. The reason why it is complex to count

Figure 1.3: Main Entrance with a possible counting line

visitors in this space is because, this is not the simple case where we have to detect a single person passing through a standard door or a straight corridor, but in opposite it is necessary to count groups of people, close to each other, walking in different directions, along a twelve meter virtual line. This case required a precise tracking system where it is possible to keep the track of the paths traversed by each individual.

Figure 1.3 shows an image of the referenced location. In the top left of the image there are the elevators and the stairwell, at the right the four meter wide main door and in the bottom of the image there are the main stairs. To the lower left of the image there are the ticket offices. To count the number and determine the direction of the people that traverse the represented line is not possible with Lasers or LEDs due to occlusions that can occur with the presence of several individuals in the area. To use a turnstile it is also not possible due to the high architectural impact restrictions imposed. In addition, the video camera is the only system capable of tracking the visitors, and it is the only available solution to count people in this area. Another problem in this particular space, if we consider that the sensors that we'll be used are cameras, is that in this area we have very bright and very dark areas. In order to have a good pixel information in every pixel of the image, the camera should possess a very good dynamic range, and therefore, have a good color information for each and every pixel.

Figure 1.4: Image of Bar dos Artistas with an example of a group of school children

Figure 1.5: Image of the main entrance with an example of a group

1.2.2 Computer Vision Tools

To engage in this task it was needed to load ourselves with computer tools that could perform image processing and in this way allow us to rapidly test different imaging algorithms. Open Source Computer Vision (OpenCV) is an open source computer vision library that runs in Linux, Windows and Mac OS X, with active development on interfaces for Python, Ruby, Matlab and other languages.

OpenCV was designed for computational efficiency and with strong emphasis on real time applications. It is written in optimized C and C++ and can take advantage of multicore processors. Further automatic optimization on Intel architectures through Intel's Integrated Performance Primitives (IPP) libraries, which consist of low-level optimized routines in many different algorithmic areas. One of the OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people building vision applications quickly. The OpenCV library contains functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision and robotics. Since computer vision and machine learning often go hand-in-hand, OpenCV also contains a full, general-purpose Machine Learning Library.

1.3 Objectives

The main goal of this thesis is to contribute to the evaluation of the possibility to develop a technical solution that allows the use of surveillance cameras for counting people in public buildings. Our contribution is in the analyses of current real time computer vision methods to create a system capable of reliably count the number of persons in complex environments. In particular, we will focus our contribution in the development of a real time people counter in complex environments, using distributed perception systems. In this way, our objectives for this dissertation are:

- Analyze the applicability of current real time computer vision methods as the solution to count the number of people that visit the building of Casa da Música do Porto;
- Test the implementation of different state of the art methods for real time still

camera background maintenance and people detection;

- Define the location of the imaging sensors that improve the reliability when counting the number of building visitors;
- Contribute with a computer vision based set of methods that allow to reliably count people in complex environments;
- Propose a system architecture for people counting using distributed perception systems for people counting systems in complex environments;
- Evaluate the proposed methods in different scenarios;
- Experimental validation of the system.

1.4 Dissertation Outline

In the next chapter is described an overview of the related work that approaches the theme of people tracking is putted together. Some of the presented work focuses, particular on the background maintenance problem, while others deal with the full spectrum of the people tracking problematic. The selected article's methods include stereo differencing and methods that use a classification schema based on pixel colors and/or textures.

The third chapter deals with the problem formulation issue, comprising the system goals set by Casa da Música Foundationand infer the system's specifications.

The fourth chapter will go deeper into the techniques used in people tracking and counting.

The fifth chapter introduces the system that was conceived.

The following chapter refers to the system implementation and concerns the issues, implied on the deployment of the conceived counting system.

The seventh chapter will be dedicated to presenting and explaining the obtained results.

On the last chapter, it will be discussed the defined goals that were achieved and the results obtained in the previous chapter.

Chapter 2

Previous work

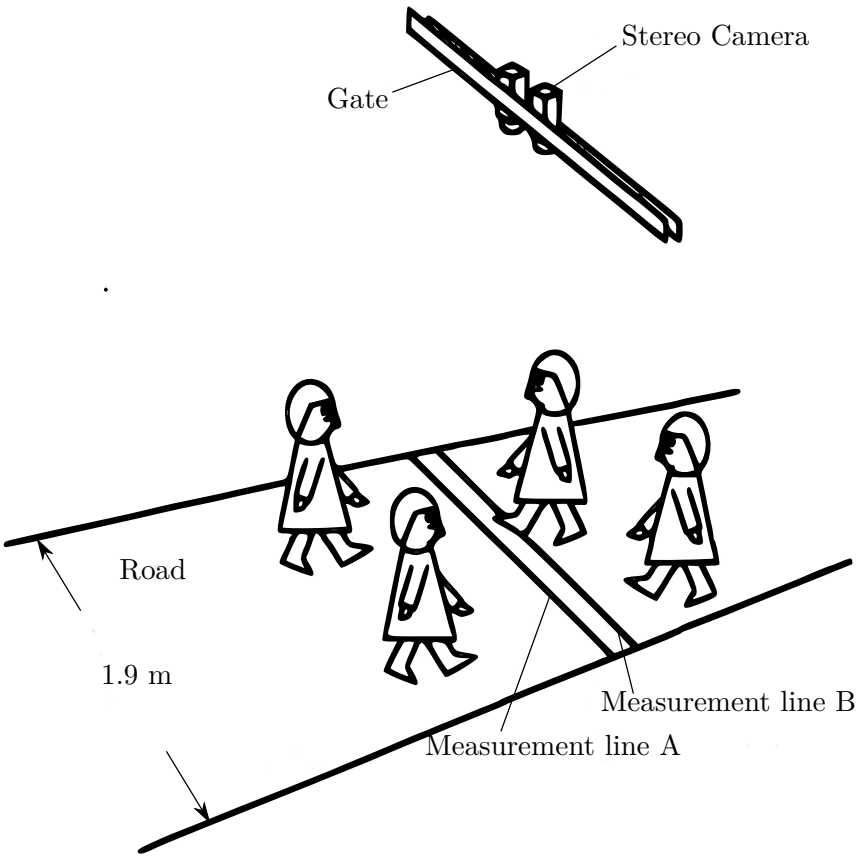


Figure 2.1: Terada's stereo camera system illustration.

To find and track people in public and/or closed spaces has been the target of a broad research. Many authors and techniques address this market necessity with more or less success. The conditions in witch the tracking takes place are of critical importance for it's success. Next we'll present some of the work closely related to the task at hands.

Using stereo differencing and an overhead camera view Terada et al. [1] created a system that can determine people direction movement and so count people as they cross a virtual line . The top-down view avoids the problem of occlusion when groups of people pass through the camera's field of view. To determine the direction of people, a space-time image is used.

Beymer also uses stereo-vision to track people [2]. Template based tracking is able to drop detection of people as they become occluded, eliminating false positives in tracking.

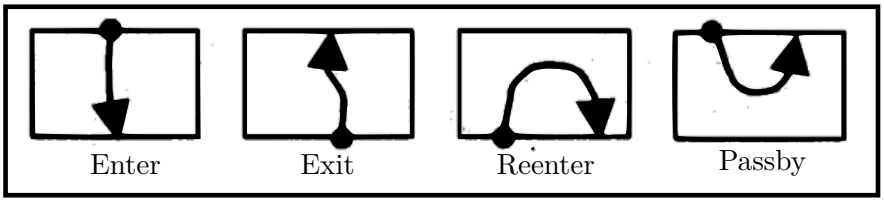


Figure 2.2: Beymer's categories for tracking classification [2].

Hashimoto et al. used a specialized imaging system designed by themselves (using IR sensitive ceramics, mechanical chopping parts and IR-transparent lenses) [3]. They developed an array based system that can count persons as they pass through a 2 meters door at a rate of 95%. In order to work in good conditions, the system requires a distance of at least 10 cm between passing people to distinguish them and thus to count them as two separate persons. Their system also shows some problem in counting with large movements from arms and legs.

Tesei et al. uses image segmentation and memory to track people and handle occlusions [4]. To extract regions of interests, their method uses background subtraction. They use the blob area, height and width, bounding box area, perimeter and mean gray level to track the blobs. By memorizing all this features over time, the algorithm cans resolve the problem of merging and separating of blobs that occurs from occlusion. When blobs merge during occlusion, a new blob is created with other features. However, this this new blob, stores the data from the blobs´s features which form it. So when the blobs separate back, the algorithm can assign the original labels to the original blobs.

Shio´s [5] work is focused around the detection of people occlusions in the background segmentation algorithm by simulating the human's perceptual grouping. First, the algorithm calculates an estimation of the motion using frame differencing and uses this data to help the background subtraction algorithm to determine the boundary between occluded persons. This method uses a probabilistic object model which has information about width, height, direction and a merging/splitting step like the seen in [4]. It was found that using an object model is a good improvement for the segmentation and a possible way to resolve the occlusions problem. But using perceptual grouping is totally ineffective in some situations like, for example, a group of people moving in the same direction at speed almost equals.

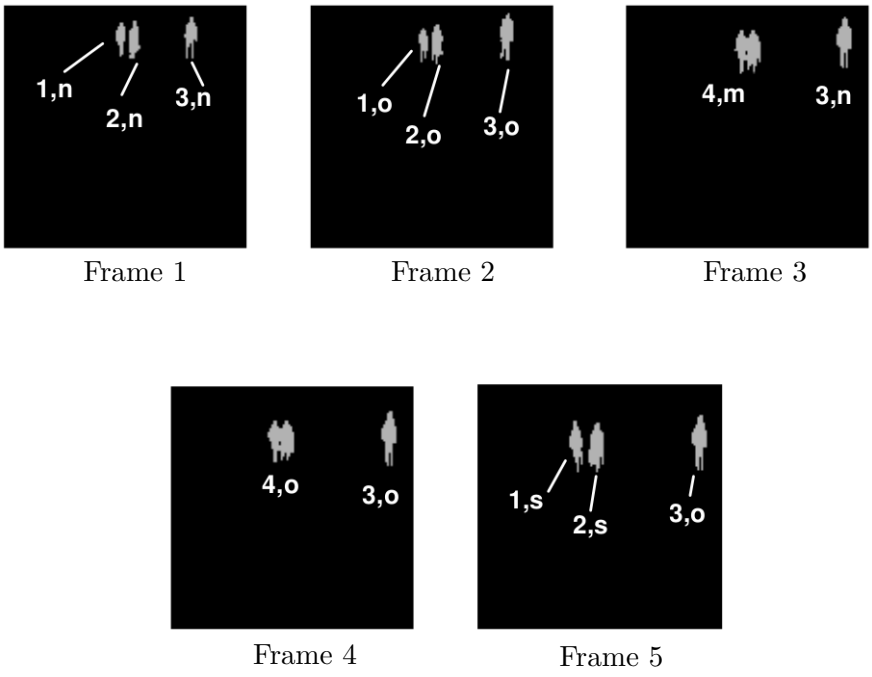


Figure 2.3: A sequence of images showing critical cases of blob splitting, merging and displacement [4].

As simple and effective approach, Sexton et al. uses a simplified segmentation algorithm [6]. They tested their system in a Parisian railway station and got a error ranging from 1% to 20%. Their system uses a simple background subtraction method to isolate people from the background, taking only the blobs centroids to make the match between frames. To improve the robustness of the system, the background model was constantly updated, reducing environmental changes impact on the system. The camera was hanged in an overhead position, also reducing occlusions and simplifying the blob detection problem.

In [7] Segen concentrates on image processing after segmentation. They extract blobs through a simple background subtraction method and then track their features between frames. The paths of each blob are stored and used to detect the intersection and its direction when crossing a virtual line. This system does´nt deal with occlusions problems, so it´s performance is greatly reduced in crowded environments. The path intersection algorithm performance was also affected negatively in crowded environments.

In [8] Haritaoglu and Flickner adopt a different method to deal with the problem of real-time tracking of people. In order to segment the foreground, they use a background subtraction based in color and pixel intensity.Pixels are classified into three different groups: foreground, background and shadow. The foreground regions are segmented into individual people by using two motion constraints: temporal and global. In order to track the individuals, the algorithm uses an appearance model based on color, edge densities and mean-shift tracker.

In [9] Gary Conrad and Richard Johnsonbaugh also use an overhead camera to simplify the problem of occlusions. In order to overcome illumination change problems, they use consecutive frames differencing instead of using background subtraction. Because of the limited computation power they had, their algorithm was designed to only act in a small rectangular area of the image. They were able to achieve a 95,6% accuracy rate over 7491 people.

In [10] Toyama et al introduces an adaptive system for background maintenance. The system is composed of three components: a pixel-level component that performs a Wiener filter make predictions of the expected background; a region-level component that fills homogeneous regions of foreground objects; and a frame level component that detects sudden, global changes in the image. Comparison of the system with other algorithms allowed them to establish five principles of background maintenance.

In [11] a new method is proposed as a improvement to the adaptive background mixture model proposed in Grimson et al [12]. That method suffered from slow learning at the beginning, especially in busy environments and from the fact that it could not distinguish between moving shadows and moving objects. The new method reinvestigates the update equations, and then uses different equations at different phases. This allowed the system to learn faster and more accurately as well as adapt effectively to changing environments. A shadow detection scheme was also introduced. It is based on a computational color space that makes use of the background model. The background Modelling uses a Expectation Maximisation (EM) algorithm to fit a Gaussian mixture model. This is more computational expensive than the previous method, but it provides the segmentation of the shadows. The shadow detector is implemented in

the RGB color space. The method used compares a non-background pixel against the current background components. If the difference in both chromatic and brightness components are within some thresholds, the pixel is considered as a shadow.

Hoprasert [13] proposed a method of detecting highlight and shadow by gathering statistics from collected images. Brightness and chromaticism distortion are used with four threshold values to classify pixels into four classes. The method that used the mean value as the reference image in [13] is not suitable for dynamic background. Furthermore, the threshold values are estimated based on the histogram of brightness distortion and chromaticism distortion with a given detection rate, and are applied to all pixels regardless of the pixel values. Therefore, it is possible to classify the darker pixel value as shadow. Furthermore, it cannot record the background history.

In [14] Jwu-Sheng proposes a new 3D cone-shape illumination model and combines a long-term color-based background model and a short-term color-based background model to solve the problems in [13].

In [15] we are presented with a privacy-preserving system for estimating the size of inhomogeneous crowds, composed of pedestrians that travel in different directions, without using explicit object segmentation or tracking. First, the crowd is segmented into components of homogeneous motion, using the mixture of dynamic textures motion model [16]. Second, a set of simple holistic features is extracted from each segmented region, and the correspondence between features and the number of people per segment is learned with Gaussian Process regression.

In [17] histogram of quantized local feature descriptors were used to represent and match tracked objects. This method has proven to be effective for object matching and classification in image retrieval applications, where descriptors can be extracted a priori. This system approaches real-time requirements.

Chapter 3

Problem Formulation

Contents

The requirements for this systems were given by the Fundação Casa da Música itself.

3.1 System Goals

Following it´s defined goals, the Casa da Música Foundationpretends to obtain an electronic people counting system to determine the total number of people that daily visit the building Casa da Música. The objective is to make available, in a quick, simple and credible way, a computer tool and a algorithm associated to the calculation of the number of visitors of the building of Casa da Música.

In order to meet these goals several issues were attained:

- For the processed information to be easily reached from within the building or even the Internet, the system should be part of a Ethernet network. At this point there is no need to create a new Ethernet network, as there are already a number of cameras installed from a previous project that aimed to create a system to satisfy the same needs. The installed cameras with the current streaming configurations did not overload the network and so it was believed that the new proposed system would not have any extra network load.

- For the system to be credible it needs to have significant accuracy. It was defined with the Casa da Música Foundationthat for the system to be considered successful it is required an accuracy up to 90%.

- For the information to be easily accessed and manipulated the system should maintain it's data history trough the use of a database. There is a Microsoft SQL database available for the diverse needs of the services inside the building and so it was idealized that the system should update this database.

- It was agreed that the source code of the application needed to be made available to the client, under a no divulge or distribution clause. This was important for the client, so that it would be easier to maintain and update the system to fulfill any future needs that could and most probably would rise.

- The system should have high availability capabilities. The system is supposed

to work unattended 24h/7d and should be able to handle events such as power failures, restarting itself and loading all configurations and calibrations, without human intervention.

- There should be a maintenance routine that checks any possible malfunctions and warns about them trough email or SMS. Since the system is supposed to work unattended and it's usage might only come trough it's stored data when, sporadicly, there is the need to make reports, situations like network failure could pass undetected for a long time.

- the cost should be lower than that of the existing systems.

3.2 Specifications

3.2.1 Technical Specifications

In the document that contained the specifications of the project, one could read about the constraints associated with this project:

The system will have to be certified by a external entity that will make sure that the estimated number of visitors does not have an error greater than 10% per counting location, for a period of thirty minutes.

The system should obtain the above results even with the following conditions:

- A large number of people will be passing by in the moments that precede and proceed concerts and other events;

- There´ll be a large number of people next to the counting zone (one of the counting zones is frequently used as a waiting area during concert´s or other events intervals);

- The illumination conditions are inconstant, either because the sun´s changing position during the day, or because of the building´s working rules;

- Installation problems coming from building's working rules, architectural, safety or technical reasons.

With these conditions in mind, videos from the locations where the control would probably be implemented were analyzed. Immediately, a critical zone was identified, where it would be technically more challenging to implement the proposed system.

The said location was the triangular shaped main entrance. With widths that could reach $14m$, it presents some of the major difficulties associated with people tracking.

1. It's geometry allows the creation of big compact groups of people with hard to segment occlusions. Those extreme situations occur mainly when spectators are waiting for a show to start or restart, or when school groups visit the building, since they are composed of tens of children walking close together.

2. At the same location there's also lighting difficulties that come from the south facing enormous glass door and window, thus allowing direct sunlight to come in the counting area. This architectural conditional causes big shadows in the floor that have to be properly segmented, while the image is too exposed in the areas close to the glass openings and under exposed in the areas further away from them.

3. That said we must not forget that this is the main entrance and associated with this locations in this type of buildings there's usually standing people, like a security worker (should he be foreground or background?) or a group of standing people waiting for an event to start or restart. This presents the classical sleeping person problem as specified by Toyama in [10].

3.2.2 Other Specifications

Another specification was that the system must be implemented in the current year, preferably within four months from the date that the project was announced due to the urgency of gathering statistical data.

Worth saying that my Master's degree is being done while having a full time teaching job. Time is of a very limited amount and represents a significant restriction to the project.

Chapter 4

Techniques and Technologies

Contents

The approach to people tracking problem can be done using different methods that can be classified into three categories:

- Methods using region tracking features. These methods use a classification schema based on pixel colors and/or textures.
- Methods using 2D appearance of humans. In these methods, the purpose of the algorithm is to match a given human model to human(s) in the image.
- Methods using multiple cameras to make a full 3D model. This approach uses more than one camera to render points of interest in a 3D reference frame.

These methods can have various degrees of complexity, however it´s possible to consider that they are displayed in increasing order of complexity, computing requirements and precision, where the last method is in the limit of today´s personal computers processing capabilities.

4.1 Tracking Category

Since we wanted to use, when possible, the existing surveillance cameras, we opted for using a traditional method using region tracking features. This method requires minimum calibration and training at startup besides being the most flexible of the three methods introduced in Chapter 1.

4.2 Camera Placement

The use of a overhead camera is generally the best option in order to avoid occlusions when groups of people pass through the camera's field of view. This placement should be preferred whenever possible.

4.3 Image processing techniques

4.3.1 Background Maintenance

In [10], Wallflower, a three component system for background maintenance, is proposed:

- the pixel level component performs Wiener filtering to make probabilistic predictions of the expected background;
- the region-level component fills in homogeneous regions of foreground objects;
- the frame-level component detects sudden, global changes in the image and swaps in better approximations of the background.

The Wiener filter is a linear predictor based on a recent history of values. Any pixel that deviates significantly from its predicted value is declared foreground. The linear prediction for a given pixel is given by:

$$s_t = -\sum_{k=1}^{p} a_k \, s_{t-k} \tag{4.1}$$

where s_t is the predicted value of the pixel at frame t, the s_{t-k} is a past value of the pixel, and the a_k are the prediction coefficients. The expected squared prediction error, $E[e_t^2]$, is

$$E[e_t^2] = E[s_t^2] + \sum_{k=1}^{p} a_k \, E[s_t \, s_{t-k}] \tag{4.2}$$

The a_k are computed from the sample covariance values of the s_n. If the actual value of the next pixel differs by more than $4.0\sqrt{E[e_t^2]}$ from its predicted value, the pixel is considered to be foreground.

This linear filter will work well with conditions like the traditional moving leaves problem in background maintenance. In [10] the filter was tested against a video sequence showing a CRT and the interference "bars" rolling up the screen with good results.

The frame level component described here is also of noticeable importance as it demonstrates a method for rapid adaptations to previously observed situations. Representative sets of the scene background models are kept and then an automatic mechanism switches between them. The decision when to change models is done by monitoring the fraction of foreground pixels in the image. If this fraction exceeds 0.7, the

consideration of a new model is triggered. The selected model is the one that minimizes the fraction of foreground pixels in the current image. The module was tested against the light switch scenario with success.

4.3.2 Shadow Segmentation

In [14], [18] and [13] we are introduced to shadow and highlights segmentations an removal. This process is of critical importance in our system, because without this process we would not be able to distinguish the shadows from the people that cast them.

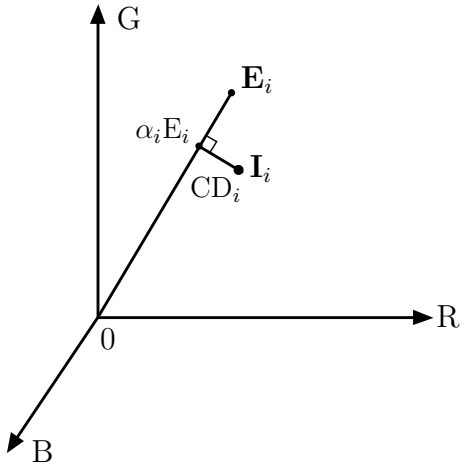


Figure 4.1: Hoprasert [13] proposed color model in the three-dimensional RGB color space

In [13] Hoprasert proposes a color model in the RGB color space that separates the brightness from the chromaticity. In this model (Figure 4.1) the background image is first modeled statistically pixel-wise. Then each i^{th} pixel can be represented by it's expected color E_i, it's standard deviation s_i and it's current color I_i . The difference between I_i and E_i is decomposed into brightness (α_i) and chromaticity CD_i components.

In [14]Jwu-Sheng Hu proposes a 3D cone, inspired by Horprasert's pillar model. This cone, combined with a Long Term Color-based Background Model (LTCBM) and a Short Term Color-based Background Model (STCBM) adds dynamic background compatibility, and a new method that calculates the threshold of brightness distortion and chromaticity distortion independently for each pixel. In [13] the darker pixels in the image were wrongly classified as shadows.

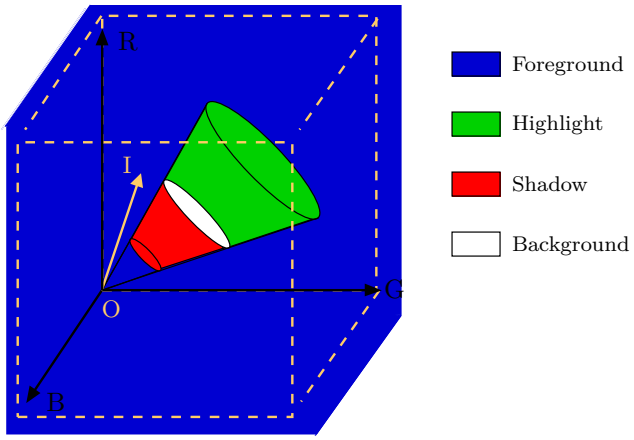


Figure 4.2: The proposed 3D cone model in the RGB color space.

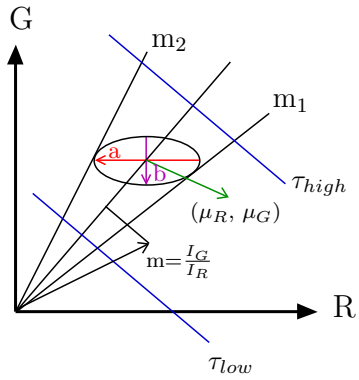


Figure 4.3: 2D projection of the 3D cone model from RGB space onto RG space.

This page was intentionally left blank.

Chapter 5

System Project

Contents

5.1 Software Architecture

The system conceived in this thesis is composed of a background estimation module that is feed with every image to build and maintain the background model. Afterwards the same image is sent to the segmentation module to build an image that contains the pixels that don't belong to the background(foreground image) and a mask that labels those pixels. In this module we also use the foreground image to find blobs, their contours, bounding boxes and centers. Finally the detections gathered in the last module are fed to the tracking and counting module to estimate the tracking and detect people crossing a virtual counting line.

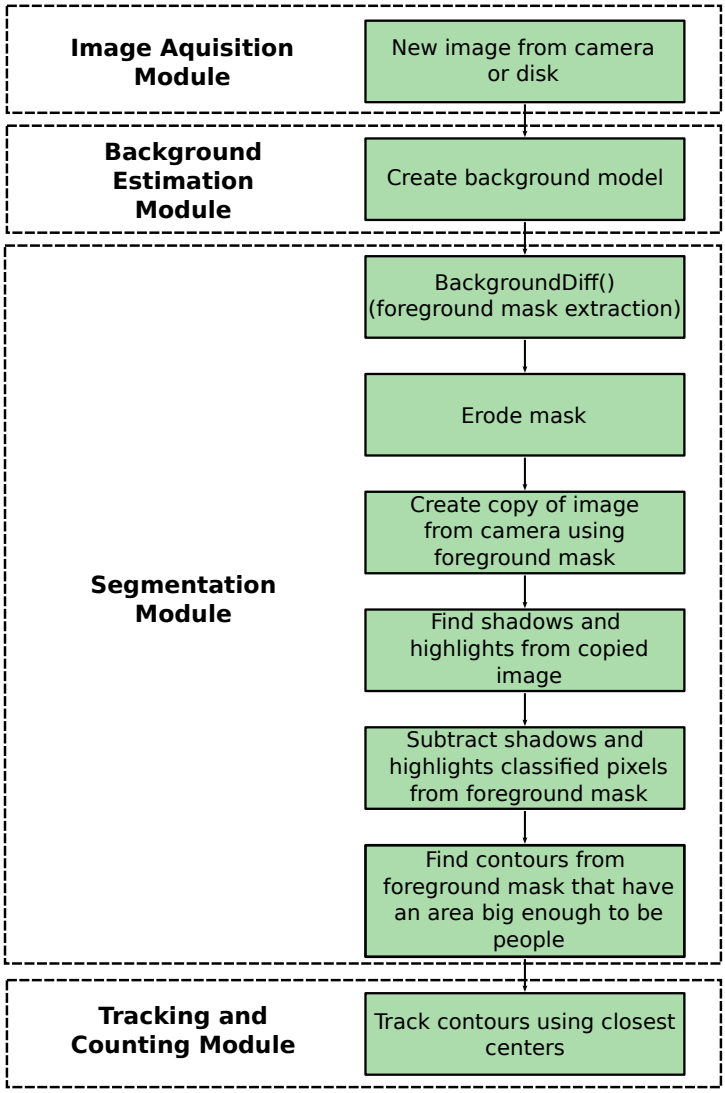


Figure 5.1: System Flow

Figure 5.2: Sample captured image

5.2 Image Acquisition

To do this work we recorded images from the security cameras in Oporto´s Casa da Música (Figure 5.2). The videos are encoded using mjpeg and are 352x288 pixels in size. Although the videos are being read from the disk, the implementation of a module that acquires images directly from the cameras is a simple task using the OpenCV libraries, since there are functions that dedicated to this task that make that process as simple as a call of a function. After that function is called, the remaining process is not affected by the option selected in the image acquisition module.

5.3 The Background Estimation Module (BEM)

This module receives the current image and builds a background average model and a background average error model. The first is accomplished by using a running average of the frame sequence. Since we are using OpenCV libraries this is accomplished with the function:

$$cvRuningAvg(\ const\ CvArr^*\ image,$$
$$CvArr^*\ acc,$$
$$double\ \alpha,$$
$$const\ CvArr^*\ mask\ =\ NULL\ )$$

The function calculates the weighted sum of the input *image* image and the accumulator *acc*, so that *acc* becomes a running average of frame sequence:

$$acc(x,y) \leftarrow (1-\alpha)\cdot acc(x,y) + \alpha \cdot image(x,y) \quad if \quad mask(x,y) = 0 \tag{5.1}$$

where α regulates the update speed (how fast the accumulator forgets about previous frames).

This function is called from within two functions that exist in this module:

- *accumulateBackground* - This simple function is used to initialize the background maintenance process. It maintains two accumulators, *IavgF* and *IdiffF*, and increments *Icount* (Figure 5.3).

- *accumulateBackgroundIf* - This more complex function is used after a predefined threshold number of frames. It does a lot more than the previous one: First, it uses the variables *IavgF* and *IdiffF*, that already have the information from the previous *Threshold* number of frames, to compute a maximum and minimum value for each pixel of each color channel. Secondly it uses this maximum and minimum values to build a mask where each pixel is set to white, if the corresponding pixel in the current frame is within this value for all color channels and set to black if it's not. The resulting mask is inverted to get a new one containing the foreground pixels. It then erodes and dilates to clean up the image (Figure 5.4), by calculating a bounding box for the blobs in foreground mask image and we are able to saturate every pixel inside that box. The resulting mask is inverted because we are looking for indexes of background pixels(Figure 5.5).

α is calculated based in the current value of *Icount*, based in the formula:

Figure 5.3: Sample image processing from Background Estimation Module: **left** - Original Image; **right** - Background Average.

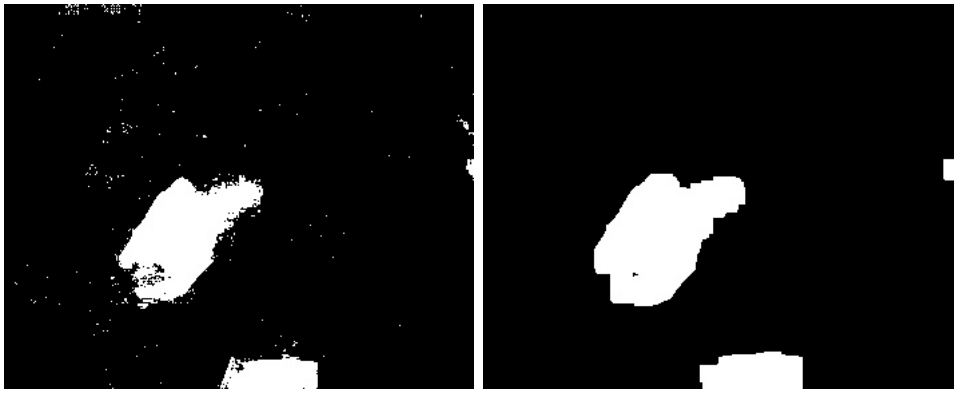


Figure 5.4: Sample image processing from Background Estimation Module: **Left** - Foreground Mask; **Right** - Foreground Mask after erode and dilate.

Figure 5.5: Sample image processing from Background Estimation Module: **Left** - Foreground Mask after Bounding Box; **right** - Background Mask (inverted previous image)

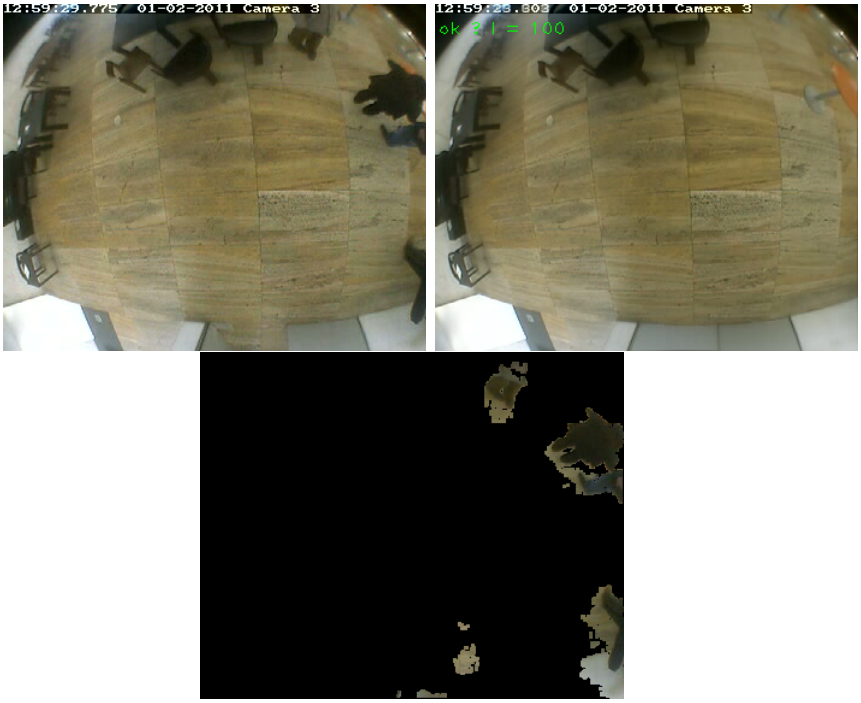
h! 

Figure 5.6: Sample Foreground Extraction: **Top Left** - Original Image; **Top Right** - Background Average; **Lower** - Foreground Image.

$$\alpha = \frac{1}{Icount} \tag{5.2}$$

In this module we call *accumulateBackground* while $Icount < Threshold$, after that we switch to the function *accumulateBackgroundIf*.

Icount is initialized with the value 1 and is incremented in every frame until it reaches the value of $Threshold$. The result is that, when we first initialize the BEM the background model is the first frame, when we process the second frame it will have a weight of 50% in the BEM, in the third frame will have a weight of 30%, etc. This way we are able to *bootstrap* the code without having any previous knowledge of the background.

5.4 The Segmentation Module (SM)

This module first calls *backgroundDiff* that uses previously computed maximals and minimals to make a mask containing the foreground pixels (Figure 5.6).

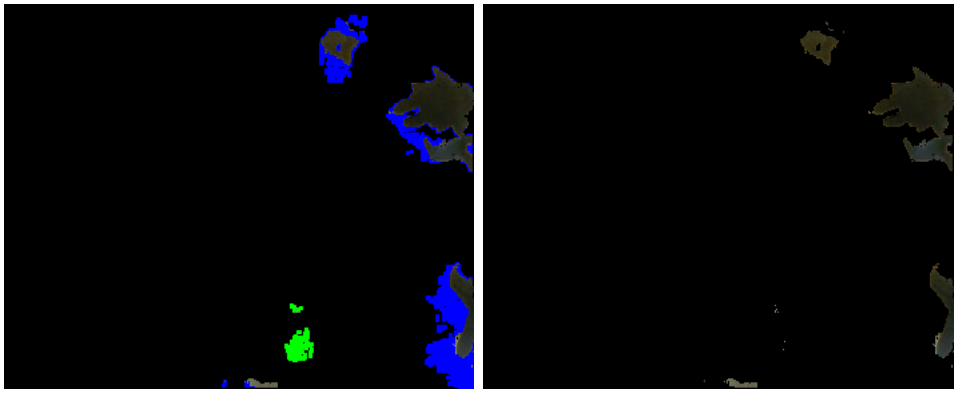


Figure 5.7: Shadow and Highlight removal: **Left** - Foreground image with pixels marked as shadow in blue and pixels marked as highlights in green; **Right** - Foreground frame after shadow and highlights removal.

The next step in this module is to erode this mask to get rid of noise. This mask is used to extract, from the current frame, a copy containing only the foreground pixels.

Using the background average and the foreground frame we estimate the shadows and highlights in the foreground image. The foreground image was built using a upper and lower threshold in the RGB color space,ie:

$$\begin{aligned} dst(I) = & lower(I)_R <= src(I)_R < upper(I)_R \wedge \\ & lower(I)_G <= src(I)_G < upper(I)_G \wedge \\ & lower(I)_B <= src(I)_B < upper(I)_B \end{aligned} \tag{5.3}$$

In this kind of thresholding, if we have illumination variations like the shadow cast from a person in the floor, we will define those shadowed pixels as foreground, when instead they are just variations of the the same color inside the 3D cone of Figure 4.2. To correctly assign those pixels to the background we need to mark them as shadows or highlights. To accomplish this segmentation we used the work developed in [14] and implemented in our function called *FindShadows* that detects shadows and highlights and marks them respectively in the two mask matrices *ImaskS* and *ImaskH*. By removing the pixels in those masks from the foreground mask we are able to get a better representation of the foreground, as seen in Figure 5.7.

Using the mask of the foreground, after the shadows and highlights removal, the contour finding routine is called. In this routine we first find the external contours of

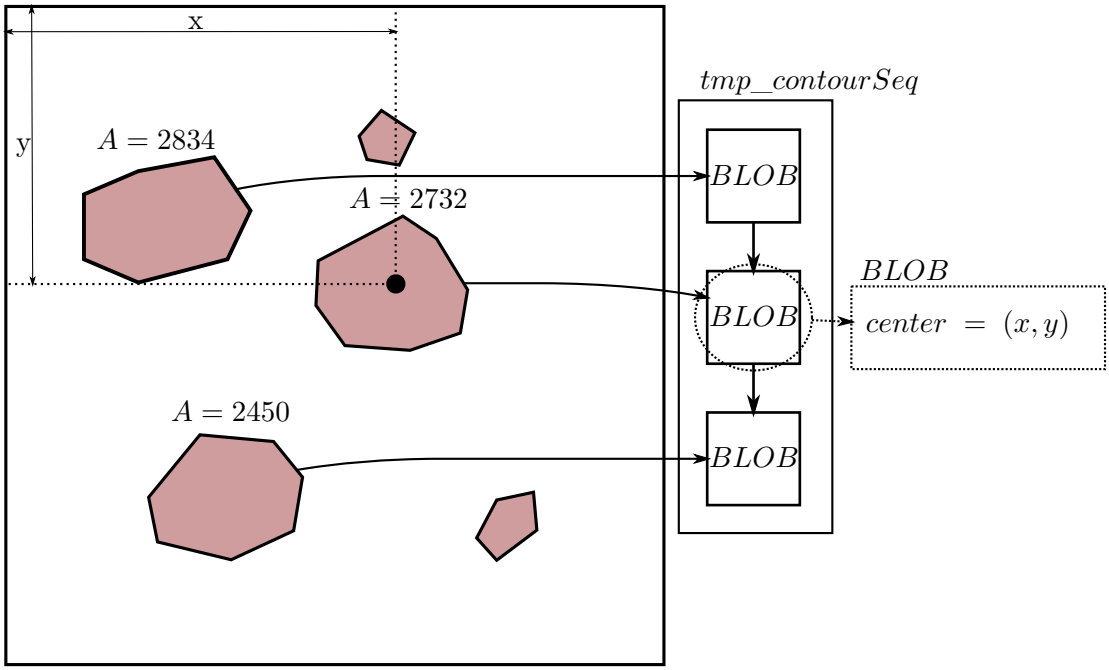


Figure 5.8: Valid blob model storage

the mask image. Then we check for each contour if they are big enough by calculating their area. If they are bigger than a predefined area then we calculate its center and save it to a structure. The structure is sent to a linked list (*tmp_contourSeq*) that will contain all detections from the current frame (Figure 5.8). In OpenCV, linked lists are manipulated through the use of the *cvSeq* type. When a *cvSeq* variable type is initialized, among other things, it receives as one of the input parameters the variable type that it will hold. In our case it's the structure BLOB.

```
typedef struct blob
{
    CvMemStorage* path_storage;
    CvSeq* path;
    CvPoint center;
    int size;
    int avrSize;
    int avrdiff;
}BLOB;
```

In this structure important metrics for the modeling of our blobs, have been created (Figure 5.8). However the system is using only the the *center* variable to model the segmented blobs. The pointers path and path_storage are used to store the path of

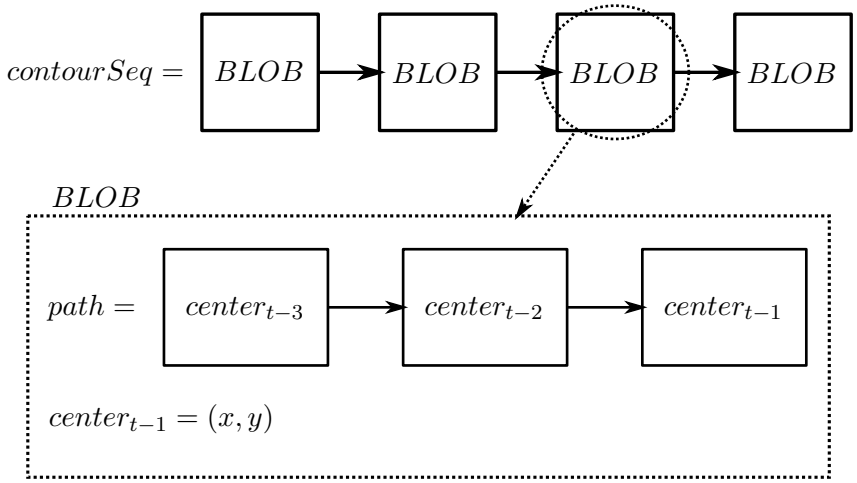


Figure 5.9: Previous contourSeq list

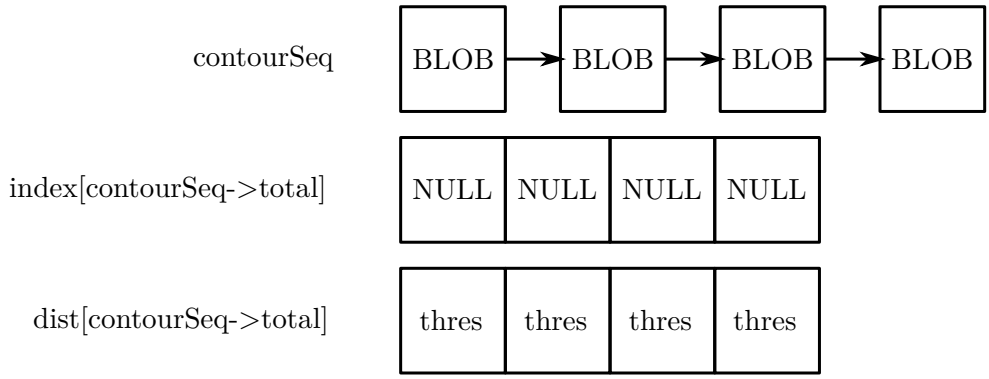


Figure 5.10: Blob association initialization

the blob in the next processing step.

5.5 The Tracking and Counting Module - TCM

This module is mainly composed of a maintenance routine that reads the linked list *(tmp_contourSeq)*, built by the previous module, and associates each element to another linked list called *contourSeq*.

The association is done by finding the minimum distance between centers of blobs in consecutive images. We also configured a threshold for the maximum distance that we consider plausible to be the same person.

In order to explain this process lets assume that the process is already running and we have the *contourSeq* variable loaded with previous detections as seen in Figure 5.9.

To associate the blobs from the current frame to the blobs in the previous frame we

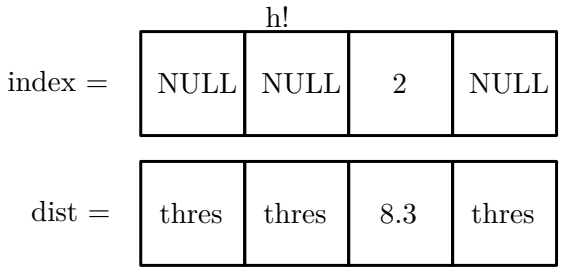


Figure 5.11: Example of blob association

use two helper arrays called, dist and index (Figure 5.10). They both have the same elements number of *contourSeq* list. All elements of index are initialized with *NULL* and the elements in *dist* with a predefined threshold (adjustable from the GUI).

Next we check all distances between the last element of the path sequence in every, previously existing, element of *contourSeq* and the newly segmented center in every element of *tmp_contourSeq*. For every calculated distance we check if it is smaller than the currently loaded distance in the corresponding position of the *dist* array. If it is, then we load it to the *dist* array, and also load the *temp_contourSeq* index that was matched, to the *index* array. For instance, lets say the innovation of the second element of *temp_contourSeq* was lower than the loaded threshold for the third element of *contourSeq*. Then we would fill the third position of the *index* array with the position of *temp_contourSeq* (number 2), and the third position of *dist* array with the measured distance (Figure 5.11).

For simplicity sake lets assume this was the only association we made. The next step is to process the information gathered with the two arrays. First we remove from *contourSeq* all elements that are filled with *NULL* in the *index* array. Secondly we add the associated blob center to the end of the *path* list of the corresponding element of *contourSeq* (Figure 5.12). Thirdly we create new elements in the *contourSeq* to hold the unassociated blobs from *temp_contourSeq* and initialize the *path*, in those new elements, with the value from the respective *center* variable

5.6 The System Interface (SI)

In order to calibrate and parameterize the system we have created a Graphical User Interface (Figure 5.13). Since we were using a Gnome Desktop the choice was between

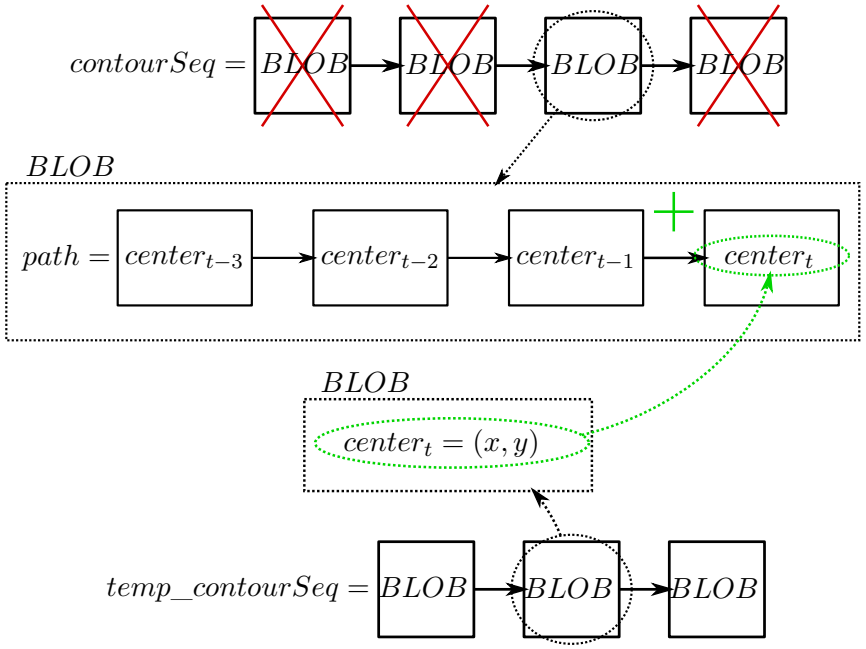


Figure 5.12: First and second contourSeq update

GTK+, the GIMP Toolkit, and Qt Framework from Nokia.

GTK+ was chosen because it's a simple system that has been widely used in our lab for several years. It's implementation seemed fast and any doubt about it's usage seemed to be easily answered by a multitude of in house people.

GTK is basically a C library for creating user interfaces. There are numerous tools to help designing the interfaces. One such tool is GLADE Interface Designer, that helps in the creation of the XML which contains the graphical interface of our application. The usage is simple. We draw the graphical interface in Glade and save the XML. This file is then loaded in the application and signals are exchanged between the graphical interface and the application's image processing loop.

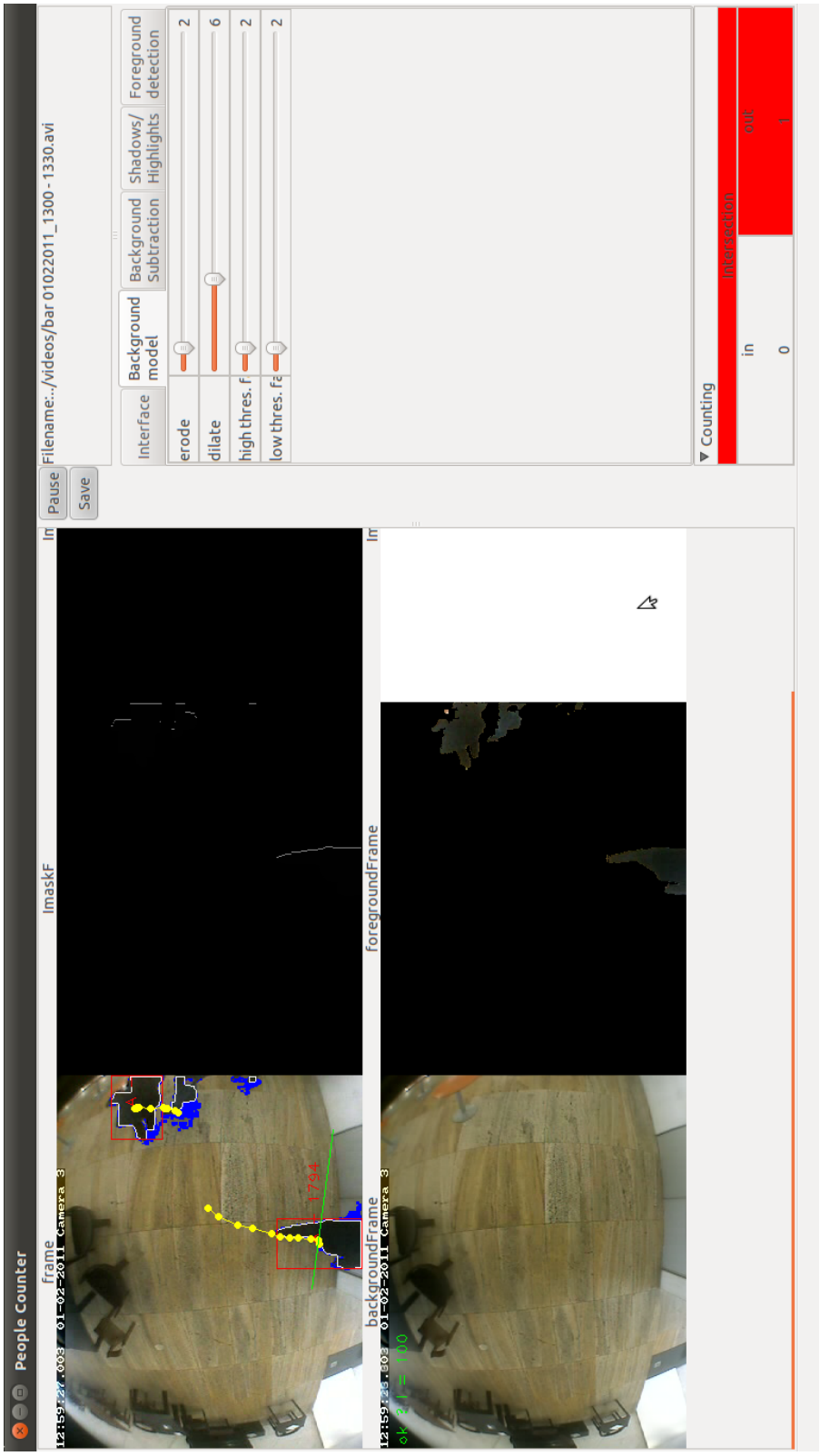


Figure 5.13: The interface developed for the tracking application

Chapter 6

System Implementation

Contents

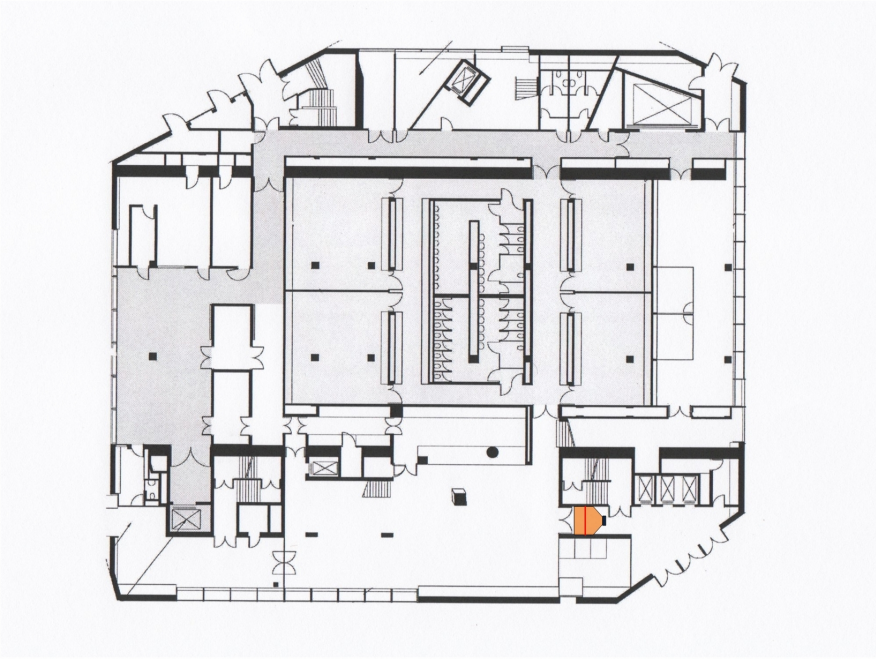


Figure 6.1: Camera implementation in the ground floor.

6.1 Camera's locations

In this section the camera's locations are presented. The system should count every building visitor, but not the people that use the underground car park. The car park, although being part of the building, has a different management.

So, to be clear about it, people are considered a visitor of Casa da Música if they use the Bar in the ground floor, or visit any of the other floors. Mind you that in the first floor, if people go from the elevators to the main entrance they should not be counted, but that we'll explain in more detail when the floor is analyzed more throughly.

Although there are other entrances in the building they should not be monitored as the people that use them are not visitors.

6.1.1 Ground floor

Close to the camera location in the ground floor, it's possible to see one of the building's exit doors, three elevators and a stairwell. The people traffic between this three locations are of no interest because when they come in or out of one of them they might be going to or coming from the car park, and therefore not a visitor of Casa da Música. Actually in this floor we can say that the only visitors are the Bar clients. In

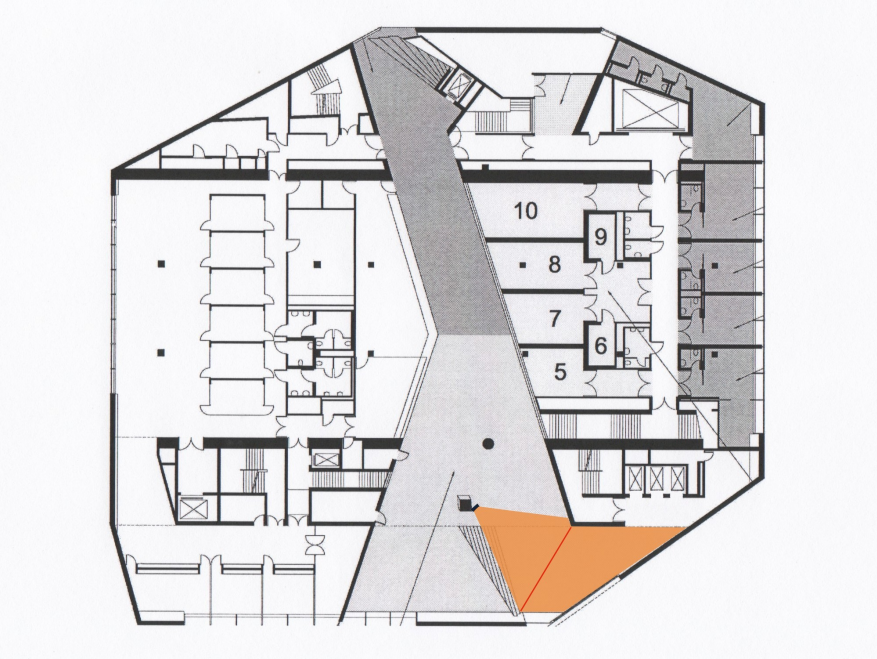


Figure 6.2: figure

Camera implementation in the first floor.

Figure 2.1 it's possible to see this's floor camera slightly facing at the door that the clients must use to enter (or exit). This camera location allows for the monitoring to be done in a straight passage and therefore improving the conditions the system works on.

6.1.2 First Floor

On the first floor we have the main entrance and the biggest challenge for our people counting system. In the blueprint shown in Figure 6.2 it's possible to see the damned counting area. The camera is placed at a height of six meters, in a $3 \times 3 \times 3$ existing metal box. that location provides a field of view of the entire counting zone. Notice the counting line, colored in red. The location chosen for it excludes the people that travel between the main entrance, the elevators and the stairwell as those people might be going to the car park. In case those people are going up in the elevators, they will then pass through other counting lines configured in the above floors.

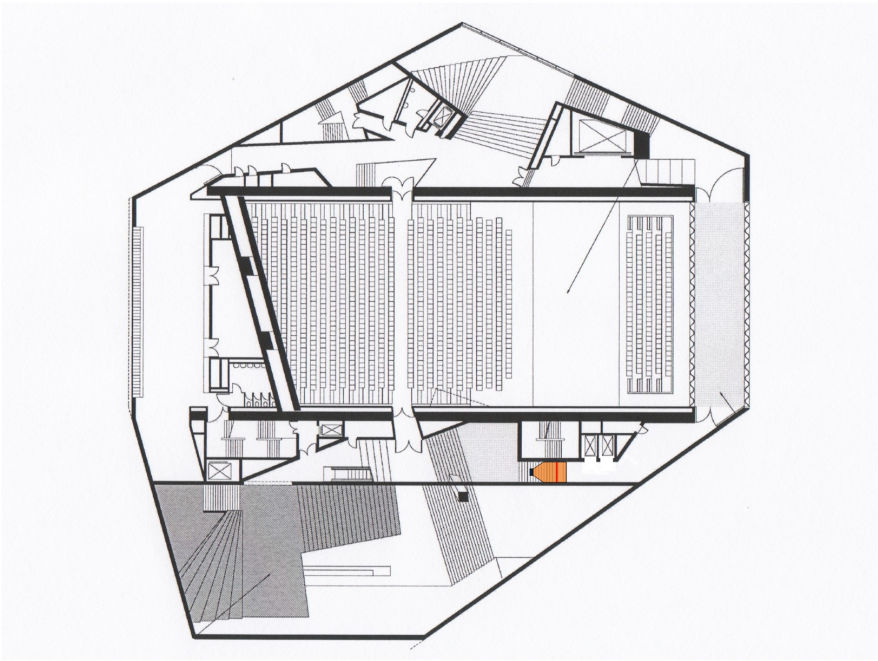


Figure 6.3: Camera implementation in the third floor.

6.1.3 Third, Fifth and Seventh Floors

From the third floor up we have the concert rooms. Visitors we'll be the spectators of the shows. The positioning of the camera is done in the corridor that accesses the elevators as seen in Figures 6.3, 6.4 and 6.5. Usually the people that come out the elevators in the third and seventh floors are spectators that have previously bought the ticket and are probably coming straight from the car park where they have left their car or from the first or ground floor in case they haven't made the course to Casa da Música by car. On the third floor, it's possible to see, in the blueprints, to the right of the elevator well, a passage to a room that has the walls to the outside of the building and to the main concert room, made of glass. Usually the doors to this glass room are closed, if this was not the case, then people could go through this room to the other floors, making it much more difficult to make a system that could account for the visitors. The all idea behind the placement of the cameras is that we are trying to count people only once per each visit they make to the building. The architectural characteristics make sure that this process is extremely difficult, and we must be able to count on some of the building's policies to be able to accurately estimate the number of visitors.

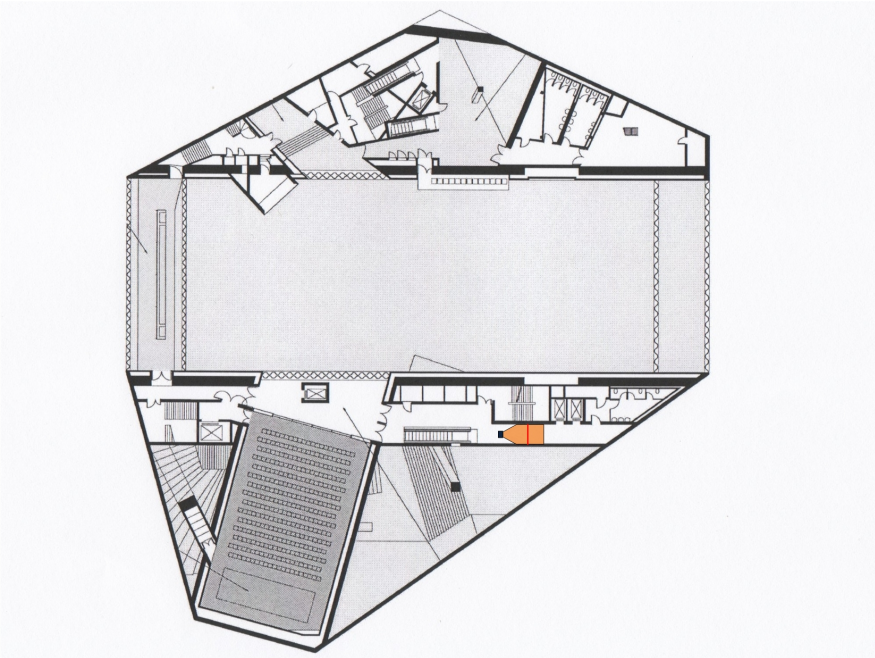


Figure 6.4: Camera implementation in the fifth floor.

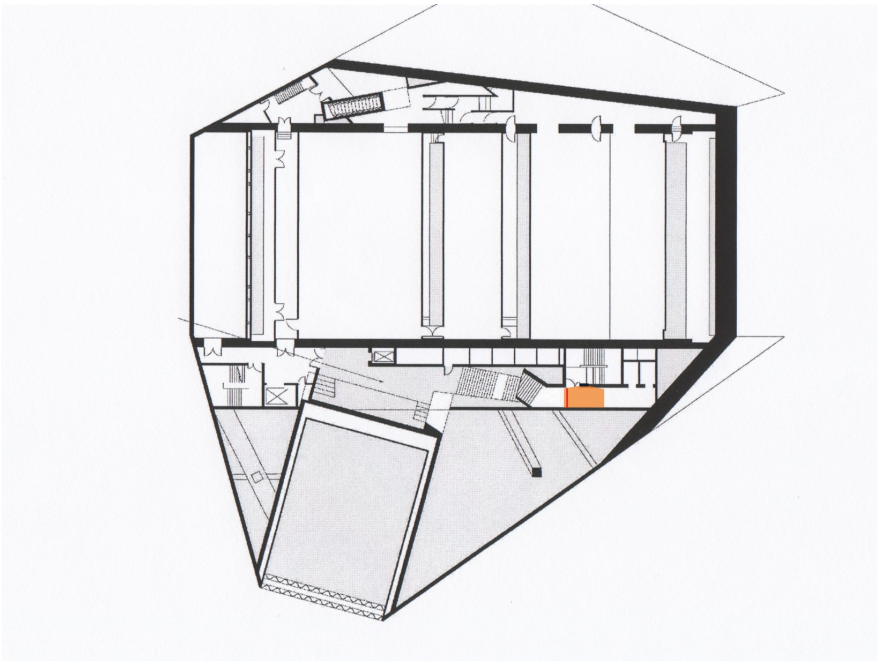


Figure 6.5: Camera implementation in the seventh floor.

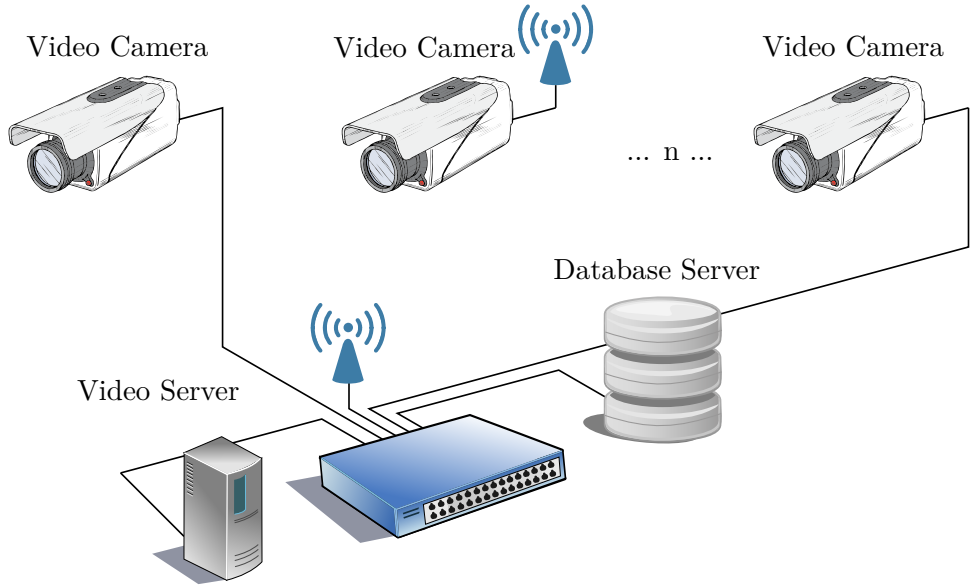


Figure 6.6: System Architecture

6.2 System Architecture

The proposed system architecture is composed of wired and wireless (where needed due to architecture restrictions) network cameras placed according to the locations illustrated in the previous section. There will be a Video Server that receives video streams from the cameras, processes it using the people counting algorithm and stores the resulting values to the database server.

Chapter 7

Results

Contents

7.1 Error Rate

In order to evaluate the robustness of the developed system, some analysis was done to a set of images recorded from the *Bar dos Artistas* in the ground floor of the Casa da Música building. The attained accuracy results were very satisfying for a first prototype, as they were very close to the desired end accuracy. This results, however, don't mean that the system is ready to be implemented in the end building, as there are some issues that need to be addressed before it can accurately count people in all previously defined situations. The main issue that needs to be solved is the problem of detecting occluded or close together people. A module for this task was not implemented as we were unable to find a process that performs the desired computation within the requested conditions. This issue poses the biggest challenge to the evolution of this system. Another issue that needs to be addressed is the global lighting variations that falsely detect background pixels as shadows due to lights being turned on or off, and due to automatic gain adjustments from the camera.

On a 10 minutes video of the bar, the system counted 13 people going in and 30 going out. The real number of people in the video are 14 going in and 34 going out, which makes an error of 7,1% and 11,8% respectively.

7.2 Background Contamination

As seen in figure 7.1, the background module has proven to have complete control over the background update process, not allowing the background to be contaminated by objects or people that stay a long time in the scene. In Figure 7.1, the background was not contaminated, even when two persons remained in the same location for more than 2000 frames. Although this module is behaving in this way, as described in section 5.3, it is definitely advantageous to add a process that checks for the time the pixels stays in the foreground, and in this way decide if they should become background. The idea is that if a table or chair is moved we end up accepting it as background and not have the background update process ignore it indefinitely.

Figure 7.1: Sample of the absence of background contamination

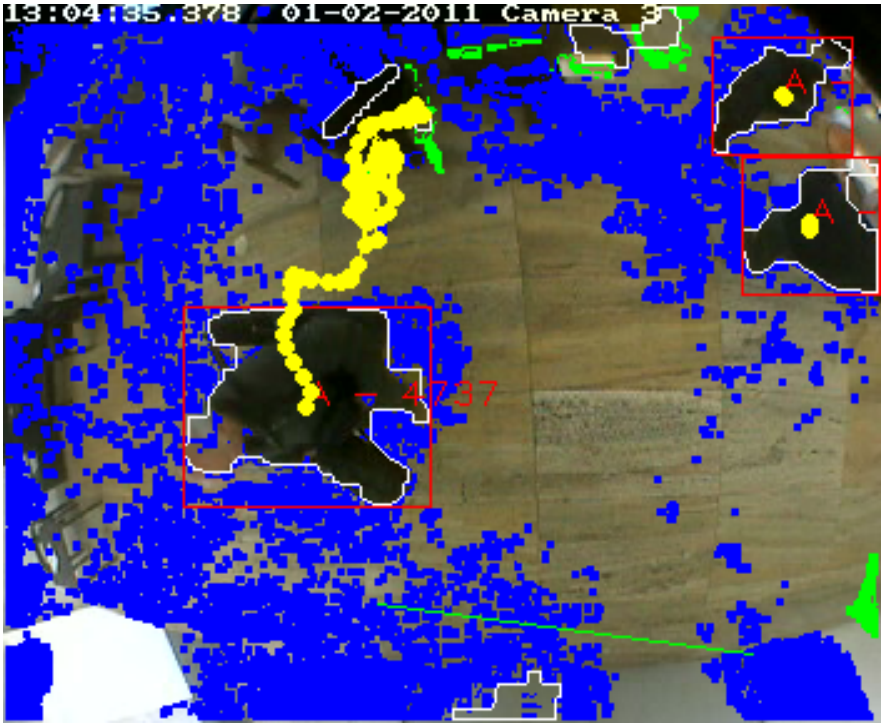


Figure 7.2: Example of shadow and highlight detection after camera auto adjusts gains

7.3 Lighting Variations

Although the system had an surprising success rate, the system is far from being complete. The next feature to develop should be a "light switch detector". Such algorithm would have to detect global light changes like the lights being turned on or off, and the camera adjusting the gains. In the tests that have been made, it's often that the camera auto compensates the gains, which creates a kind of "global shadow" or "global highlight" to be cast over the whole image. Figure 7.2 is a image taken from a moment after the camera had just adjusted the gain, and where it's possible to see a predominance of blue marked pixels. This color means that the pixel was detected as being a shadow.

To overcome this background maintenance problem we would need to create a module similar to the Long Term Color-Based Background Model (LTCBM) presented in [14]. In the referred article Hu uses a module to propose an alternate background model that tries to better model the global light changes that happen for instance, when the room lights are turned on or off.

Figure 7.3: Example of what happens when there is an occlusion.

7.4 Occlusion/Clustering Detection

The system, although having a low error in the video where the manual counting was done, was prone to fail when people come too close to each other.

More particularly, for the case in the image, contours around the two people in the image, come too close to each other, becoming a single contour. Different methods exist that address this problem.

One possibility to "solve" this problem would be to estimate the number of people of each blob by keeping a history of the blobs different metrics like what was done in [4]. With that module the system would then be better prepared to keep track of the blobs while they merge and separate.

Another much more critical case of occlusion is when we have, instead of two people coming close together, we have larger groups, like in Figure 1.4. Although it might be possible to estimate the number of people based in the size of the blob, in order to count accurately the people that traverse the virtual counting line we need to have a method that accurately segments each person inside the blob, and for that we haven't

found a real-time method.

Chapter 8

Conclusion

This dissertation evaluates the existing techniques and technologies to create a product capable of using distributed perception systems to count people in complex environments. It was concluded that it is possible, up to a certain degree of occupation density, using state of the art solutions, to count (and track) people in complex environments, although none of the existing methods in the state of the art allows a precise people segmentation (and needed tracking) when the blobs represent a large number of people.

Different state of the art methods were implemented and tested for a still camera background maintenance and people detection was performed throughout the development of an algorithm, which resulting modules have been put to more thorough scrutiny, with the results described in chapter 7.

In order to support the system development, a graphical user interface was built to tune the parameters involved in the proposed algorithm. Additionally, video control is also possible, allowing to pause each video for a better analysis of specific scenes and instants.

The modules robustness was verified, considering the limitations discussed in chapter 7. After the implementation of further methods stated in chapter 2, the system would allow the adaptation of the background module to work well in all identified scenarios. In spite of this, the segmentation and tracking module was always bound to fail in high density, crowded scenarios. The unique solution to accurately detect people in crowded scenarios, would be using more complex non real time image processing methods, like optical flow or mixture of dynamic textures.

The locations of the imaging sensors were defined and, to improve reliability, it is crucial to select an overhead pose to minimize occlusions. The overhead pose should also avoid situations where shorter persons become totally hidden by taller ones, having at all times a view of at least part of the individual. In the implemented system, since we are not dealing with occlusions, it becomes even more important to reduce them with the camera pose, in order to minimize the system counting error.

We have accomplished to contribute and present the system described in chapter 6, by putting together a set of studied computer vision methods and by slightly adapt-

ing and adjusting them, allowing to reliably count people in complex environments. Although, tests have proven that the presented methods are not appropriated for the proposed scenario, but undoubtedly adequate for other, less crowded, environments.

The distributed perception system architecture proposed in section 6.2, although not tested in the final location, allows the system to be scalable and adaptable to different scenarios. The limitations of this architecture are that all the processing power is in the video server, which makes it directly related to the number of image sensors that can be applied to the system. This number can easily, be increased by lowering the resolution or the encoding of the streamed video. However, it should be referred that capturing video from a network stream was not tested, neither was the load on the network, but its viability is assured since the cameras on the locations are already streaming to the network without overloading it. In addition, the gathered location videos were taken from these streams and have proven to have sufficient quality for the task at hands.

The tests preformed with recorded images from the locations, enabled us to determine the applicability of the algorithms in all of them, as far as the environment is not overcrowded. Also, we have determined that for the main entrance (Figure 1.3), due to the big lighting variations, it is required to have a camera with a high dynamic range, since the installed camera is lacking this feature and the color information becomes scarce in the areas infected by high lighting deviations.

For future work the system should continue to evolve as previously discussed in chapter 7, by improving the background module, to be more robust lighting wise, and adding a blob segmentation module, that is able to either segment people that are inside the same blob or a system that is able to track and maintain a history, about the characteristics of the blobs when they merge and separate, to keep track of information about individuals and sets of individuals. The interface can also be greatly expanded, for instance, by creating new check boxes that are able to turn modules on or off according to the circumstances. Such functionality is also a big help when testing the results of a new module or when comparing two similar modules (one can verify the applicability of two similar background modules in different scenarios, by using a radio button to switch between them). As a future task, for the addressed scenario in this

dissertation, and due to the difficulty to count people with RGB cameras in crowded environments, the problem mentioned should be solved using a depth camera, allowing to map the scene in 3D space besides the RGB. The depth camera works with infrared light, becoming useless in direct sunlight. For the main entrance the location of such camera would have to be chosen in a way that it captures the side of the persons that is not directly exposed to the sun and thus invisible to the depth camera.

Bibliography

[1] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "A method of counting the passing people by using the stereo images," in *Proc. Int. Conf. Image Processing ICIP 99*, vol. 2, 1999, pp. 338–342.

[2] D. Beymer, "Person counting using stereo," in *Proc. Workshop Human Motion*, 2000, pp. 127–133.

[3] K. Hashimoto, K. Morinaka, N. Yoshiike, and S. Kawaguchi, C.and Matsueda, "People count system using multi-sensing application," in *Proc. Int Solid State Sensors and Actuators TRANSDUCERS '97 Chicago.Conf*, vol. 2, 1997, pp. 1291–1294.

[4] A. T. A. T. C. S. R. G. Vernazza, "Long-memory matching of interacting complex objects from real image sequences," in *no book*, 1996.

[5] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proc. IEEE Workshop Visual Motion*, 1991, pp. 325–332.

[6] G. Sexton, X. Zhang, G. Redpath, and D. Greaves, "Advances in automated pedestrian counting," in *Proc. European Convention Security and Detection*, 1995, pp. 106–110.

[7] J. Segen, "A camera-based system for tracking people in real time," in *Proc. 13th Int Pattern Recognition Conf*, vol. 3, 1996, pp. 63–67.

[8] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2001*, vol. 1, 2001.

[9] G. Conrad and R. Johnsonbaugh, "A real-time people counter," in *Proceedings of the 1994 ACM symposium on Applied computing*, ser. SAC '94. New York, NY, USA: ACM, 1994, pp. 20–24. [Online]. Available: http://doi.acm.org/10.1145/326619.326649

[10] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proc. Seventh IEEE Int Computer Vision Conf. The*, vol. 1, 1999, pp. 255–261.

[11] P. AaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *In: Proc. AVBS 01*, 2001.

[12] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Computer Society Conf Computer Vision and Pattern Recognition*, vol. 2, 1999.

[13] *A statistical approach for real-time robust background subtraction and shadow detection*, 1999.

[14] J.-S. Hu, T.-M. Su, and S.-C. Jeng, "Robust background subtraction with shadow and highlight removal for indoor surveillance," in *Proc. IEEE/RSJ Int Intelligent Robots and Systems Conf*, 2006, pp. 4545–4550.

[15] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2008*, 2008, pp. 1–7.

[16] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," vol. 30, no. 5, pp. 909–926, 2008.

[17] L. F. Teixeira and L. Corte-Real, "Video object matching across multiple independent views using local descriptors and adaptive learning," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 157–167, 2009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167865508001232

[18] G. Monteiro, J. Marcos, M. Ribeiro, and J. Batista, "Robust segmentation for outdoor traffic surveillance," in *Proc. 15th IEEE Int. Conf. Image Processing ICIP 2008*, 2008, pp. 2652–2655.