FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Human detection solution for a retail store environment

Vítor Joel do Nascimento Araújo

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Jaime Cardoso, PhD (FEUP)

Co-Supervisor: Pedro Carvalho, PhD (INESC Porto)

February 28, 2014

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES 2013/2014

A Dissertação intitulada

"Human Detection Solution for a Retail Store Environment"

foi aprovada em provas realizadas em 07-02-2014

o júri

Presidente Professora Doutora Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Manuel Castro Torres
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade Fernando Pessoa

Professor Doutor Jaime dos Santos Cardoso
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Doutor Pedro Caryalho
Investigador do INESC - TEC

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Vítor Joel do Nascimento Araújo

Faculdade de Engenharia da Universidade do Porto

Resumo

Apesar de se terem vindo a tornar mais comuns, os sistemas de deteção humana ainda estão longe de ser perfeitos. Uma área particularmente pouco explorada é a deteção em imagens de baixa qualidade, já que a maior parte da investigação realizada na área da deteção humana do campo de visão por computador é realizada em sequências com um bom nível de qualidade, geralmente filmadas em ambientes bem iluminados, de forma a alcançar os melhores resultados possíveis. No entanto, isto cria uma falha nos atuais sistemas de deteção, já que este tipo de ambientes não corresponde a uma grande parte das gravações realizadas no dia a dia.

Esta dissertação tenciona apresentar um método de deteção que lide com os problemas de sistemas de vigilância típicos, particularmente em ambientes de loja de retalho. Nestes locais o sistema de vigilância não é, geralmente, um investimento prioritário, o que resulta na gravação de ficheiros de vídeo com baixo nível de detalhes e com elevadas taxas de compressão que geram uma grande quantidade de artefactos de codificação. Os objetos presentes nestes ambientes também tendem a causar ocultação parcial de pessoas, tornando o trabalho dos algoritmos de deteção ainda mais complexo.

A solução final melhora os resultados de algoritmos de estado da arte, sendo configurável e flexível para ser adaptada para outras sequências de vídeo. O potencial para integração com mais *frameworks* de deteção está também presente, fazendo desta solução um bom ponto de partida para outras soluções integradas de deteção.

Abstract

Even though human detection systems are becoming more common, their level of accuracy is still far from being perfect. A particularly under-explored target are low quality images, as most research done in the human detection area of the computer vision field uses sequences with a good image quality, usually filmed in well lit environments, in order to achieve the best possible detection rates. However, this also creates a gap in the current human detection frameworks, as these environments fail to emulate a large amount of real world video recordings.

This dissertation aims to provide a detection method that deals with the drawbacks of typical surveillance systems, particularly those in retail store environments. In these locations, the surveillance system is usually not an investment priority, which results in the stored video files to present a low amount of details and high compression rates that generate a large amount of encoding artifacts. The cluttered environments can also tend to easily cause partial occultation in a large amount of subjects, making the work of the detection algorithms more complex.

The end solution improves the results from state of the art algorithms, while being configurable and flexible to be adapted for other video sequences. The potential for an integration with more detection frameworks is also present, making this solution a good starting point for other human detection pipelines.

Acknowledgements

The work developed in this dissertation would not be possible without the help and contribution from some people. I take this moment to thank them for their support.

To Prof. Dr. Jaime S. Cardoso and Dr. Pedro M. Carvalho I thank for the help and guidance provided during the course of this dissertation.

I thank my parents, for bestowing upon me the opportunity to proceed my studies, and for the support and patience they have had over the years.

I also thank my friends, for supporting me through the better and worse times, and providing me good moments of joy and relaxing from work.

Vítor Araújo

"The fact that an opinion has been widely held is no evidence whatever that it is not utterly absurd; indeed in view of the silliness of the majority of mankind, a widespread belief is more likely to be foolish than sensible."

Bertrand Russell

Contents

List of Figures

List of Tables

Symbols and Abbreviations

Acronyms

1D	1 Dimension
CENTRIST	Census Transform Histogram
CPU	Central Processing Unit
CT	Census Transform
CVML	Computer Vision Markup Language
DR	Detection Rate
FAR	False Alarm Rate
FBI	Federal Bureau of Investigation
FN	false negative
FP	false positive
FPPW	false positives per window
FPS	frames per second
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
Inria	Institut National de Recherche en Informatique et en Automatique
LBP	Local Binary Pattern
MIT	Massachusetts Institute of Technology
MOG	Mixture of Gaussians
RGB	Red Green and Blue color model
SIFT	Scale Invariant Feature Transform
SR	super-resolution
SVM	Support Vector Machine
TF	total frames
TN	true negative
TP	true positive
VGA	Video Graphics Array

Chapter 1

Introduction

This chapter contextualizes the current importance of automatic human detection systems and their major advantages and weaknesses. It characterizes the detection environment targeted by the solution developed during this dissertation, the main objectives and the structure of the rest of the document.

1.1 Motivation

In recent years, the spread of video surveillance systems to the masses brought with it the desire to have new uses for the acquired data. One of those uses is human detection and tracking.

By employing detection and tracking software, surveillance systems have the potential to become even more useful: police authorities could use the data from video surveillance cameras to automatically identify a suspect of a crime, or a missing person; retail store owners could track potential costumers in order to optimize their shopping experience and help boosting sales; marketers could use it to detect when a person is passing by a screen that displays advertisements, estimate their age, sex and ethnicity and deliver targeted ads [8]. However, the current state of the art in human detection is still far from perfect. Image recognition software has been the subject of many improvements over the years, but still has a long way to go before it achieves the same quality of detection that a human is capable of. For instance, in the 2013 Boston Marathon bombings in the United States of America, an unprecedented amount of footage had been recorded. This footage was then process by the authorities, but the results came up empty, despite the fact that both suspects that were being looked for were already in the FBI database [9]. The suspects ended up being detected by a human that was looking through the video. The lack of effectiveness of the system was attributed to several causes, like the low resolution of the cameras and the long range of the recordings, some of which were badly focused and caught from angles that fell within the software weak points. As such, a need for better detection solutions is a constant presence in this field.

Current detection systems focus on good quality footage, as the main objective during their development is achieving the best detection rates. This lack of focus on low quality images makes

most detection frameworks perform poorly on this kind of footage. The algorithms either cannot point out most of the possible detections, or come up with a large amount of false positive results.

This is the area targeted by this dissertation. The solution presented here will improve the results achieved by current state of the art algorithms, while being flexible to be adapted and integrated with more detection frameworks.

Despite the continuous evolution of video recording systems, with high resolutions and good low light performance, the currently deployed solutions should not be forgotten, as the adoption of higher quality systems will not happen overnight. The large costs associated with processing and storing high quality footage make it impossible for most stores to deploy them as their surveillance system. As such, lower quality systems, with computationally low capabilities and small storage solutions, will remain prevalent for years to come, making the work developed in this dissertation the more relevant for human detection in a large portion of retail stores.

1.2 Retail environment

As the solution developed in this dissertation is targeted at human detection on a retail store, the particular challenges that this kind of environments present to human detection systems should be noted.

Retail store environments tend to present a wide variation of lighting conditions, from very dark to very bright. Sometimes, these variations can even be seen within the same store (fig. 1.1a). This problem is not so common in office environments, where the illumination is usually more even across the whole area (fig. 1.1b).

The presence of a variety of objects in stores also difficults the detection process, as areas around the subjects in the scene are usually cluttered, with elements comprising a wide range of sizes, and sometimes a large number of people are also present in the same area. These problems are not as relevant in other environments, such as some streets, where there is more open space available and a lesser change of occultation (fig. 1.1c).

Lastly, as described earlier, the surveillance systems in these locations are usually of very low quality. To process and storage a large number of recorded hours from multiple cameras would require a large investment that most retail stores cannot afford. As such, and despite the continuous improvements in technological solutions, the footage obtained from these systems tends to present a low level of details, with a large amount of compression artifacts and a low dynamic range (fig. 1.1a).

1.3 Objectives

The main objective of this dissertation was to develop a human detection solution to be applied on video sequences of the same nature as the ones described in section 1.2, and it should be able to provide better detection results than what is possible with current state of the art approaches.

For this purpose, a combination of preprocessing methods, existing detection algorithms and solutions, and post-processing approaches was tested and some used. The end solution is also adaptable to other target sequences, by changing certain configuration parameters, in order to account for the particular challenges that each video recording environment presents.

1.4 Document structure

This document is structured into chapters, each referring to the themes that follow.

In chapter 2, a state of the art analysis and related work is presented, along with results from implementations that may serve as a basis for the current work.

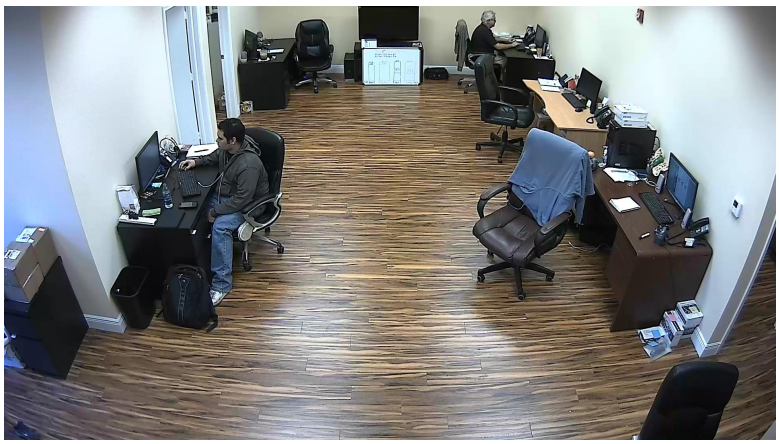
Next, chapter 3 reveals the preparatory work that occurred before the solution development could begin, such as sequence recording and annotation, and metrics selection.

Chapter 4 describes the stage of preprocessing the image sequences before the detection. This chapter presents the methods tested and employed to improve the results, along with the detection algorithms that are used in the solution developed.

In chapter 5, the workflow of final solution is described, along with its results.

And finally, in chapter 6, the author ponders on the results achieved, challenges faced and suggests paths for improving the solution developed.

(a) Retail store surveillance sample



(b) Office surveillance sample [10]

(c) Street surveillance sample [11]

Figure 1.1: Samples from different surveillance environments

Chapter 2

State of the Art

Presented here is an overview of the current state of the human detection field, some of the available detection frameworks and image processing methods that are relevant to the work being done in this dissertation. In the end, the solution developed should take advantage of the work presented here, while also providing better results than what is possible with the current state of the art.

2.1 Foreground detection

Foreground detection, or background subtraction, is one of the methods that can be used during the preprocessing stage, before the detection is done. During this process, a binary image (the foreground mask) containing all the elements from the scene that are not part of the fixed background is generated. This mask can then be applied to the source image, leaving only these elements for the detection algorithms to process, helping to reduce false positives. However, the masking has to be done in a precise manner, bearing the risk of removing elements that should be part of the foreground, and possibly eliminating human subjects from the image. On the other hand if the masking is too broad, too many background elements could be left in the final image, diminishing the effectiveness of this process.

One of the methods available is the Mixture of Gaussians (MOG) [12] [1]. This approach takes a sequence of images, generates a background model and then creates the mask for each image by subtracting the model from the image. The model is then updated at a defined interval, to account for changes in the scene (like waves in an ocean, or objects being moved in a store). For computing the background model, a cluster of pixel values from the images that serve as a source for the current model is created. Then, the maximum likelihood value is calculated according to a distribution of probability and used for the background model. This method produces a well defined foreground mask on images with good quality and illumination (fig. 2.1).

A median filter can also be used for computing the background image from a group of images [13]. By calculating the median value for each pixel position from those images, the background image is generated and can then be subtracted from the sequence as in the MOG method.

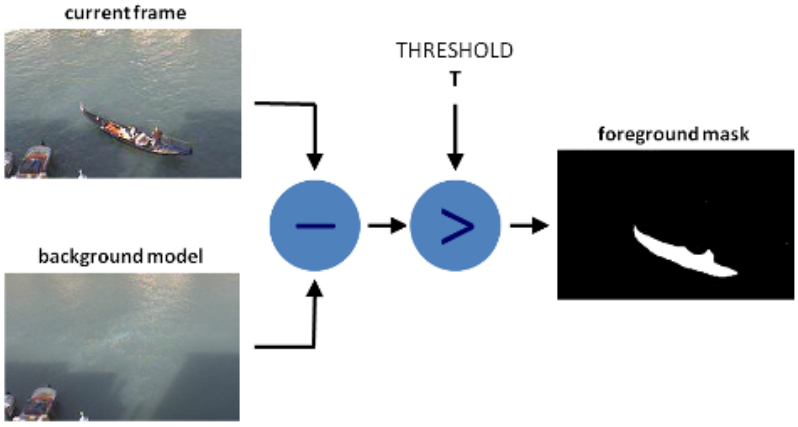


Figure 2.1: Background subtraction process [1]

Some configuration parameters can usually be fine tuned when using these methods, if the implementation allows it, making it possible to obtain better results with the dataset in use. The values that can typically be changed are the number of images used for the background model calculation, the number of skipped frames between each sample and the rate at which that model is updated. For higher frame rate videos or image sequences a lower update rate can be used, as the variation between each image is not very high. On the other hand, as the subjects will stay in the same location in more frames than for lower frame rate videos, the number of images to use for each update should be higher or more spaced, or else parts of the foreground elements could be considered by the algorithm to be part of the background.

These methods can have a wide variation in their output, depending of the scene conditions and image quality. If the lighting keeps changing in the scene, the configuration variables should take that into account, making the background update more frequently. The presence of encoding artifacts can also affect the background image creation, as individual pixel values can change drastically between frames, even if there is no movement in that area.

Despite the apparent effectiveness of these methods, their implementation needs to be carefully planed, and their configuration should be personalized for the type of footage that is being dealt with.

2.2 Feature extraction

The detection of particular elements in an image starts with the selection and extraction of features that allow for a correct identification of those elements, while reducing the amount of input data to be processed later. Within the human detection field, the following descriptors that characterize the features to be extracted can be used.

2.2.1 Histogram of Oriented Gradients (HOG)

The concept behind HOG descriptors is that the distribution of edge directions, or intensity gradients, can define the appearance and shape of local objects [3].

The implementation of this descriptor is done by dividing the image into spatial regions, called cells, which will contain a local 1D histogram of gradient directions, or edge orientations, over the cell pixels. The combination of histograms from all cells forms the image descriptor. In order to minimize the effect of illumination and shadowing variance, the local results should be contrast-normalize. This can be achieved by measuring the intensity across a larger area, called block, and using this value to normalize all cells within the block.

The major flaw that has been pointed in this kind of descriptors is their low discriminative power [14] [15]. Being based on gradient directions makes the histogram computation susceptible to complex backgrounds, which difficults the feature extraction process. Another drawback of this method is its low computational efficiency, which can be improved by using a cascade of detectors [16].

2.2.2 Local Binary Pattern (LBP)

The LBP descriptor works by first dividing the image into cells, and each pixel in a cell is compared to its 8 neighbors. If the value of the center pixel is higher than that of the neighbor, the neighbor value is replaced with a 1. Otherwise, it is replaced with a 0. Following a clockwise or counter-clockwise count, an 8 digit binary number can be generated in each comparison group. Computing the histogram of all these 8 digit numbers in each cell and concatenating all the histograms results in the feature vector that characterizes that image [17].

The LBP descriptor performs particularly well in texture detection [18] [19]. Combined with its capability to filter out noises [20], makes it a good combination with HOG [21], as the histogram-based descriptor is weaker when the background is filled with noisy edges.

2.2.3 Census Transform Histogram (CENTRIST)

This descriptor first appeared in the paper *CENTRIST: A Visual Descriptor for Scene Categorization* [2] in 2009. As defined in this paper,

> "Census Transform (CT) is a non-parametric local transform originally designed for establishing correspondence between local patches [22]. Census transform compares the intensity value of a pixel with its eight neighboring pixels (...). If the center pixel is bigger than (or equal to) one of its neighbors, a bit 1 is set in the corresponding location. Otherwise a bit 0 is set." [2]

The output is a 3x3 binary matrix, with the central position empty, which can be translated into a base-10 number in the [0 255] range. This number represents the Census Transform value for the central pixel. By repeating the process for each pixel in the image, the resulting set of values

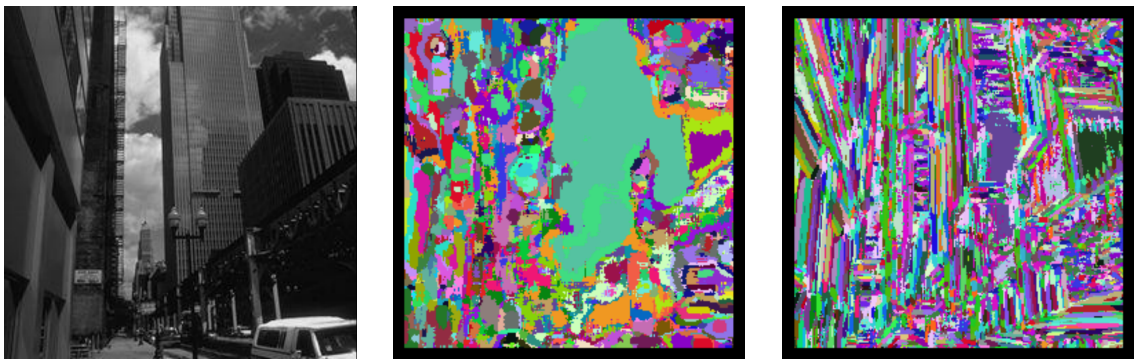
can then be used as input to the classifier. While the binary matrix generation method is similar to the one employed by the LBP descriptor, by applying the base-10 number to the central pixel, instead of using it in a local histogram, a relation can be established between neighboring pixels.

The initial processing of the image using the Census Transform method allows the classifier to do an easier recognition of the scene, as it can ignore distracting elements, like textures and color, and focus on the more important geometric features and structural properties [2] (fig. 2.2b), which are also the most relevant characteristics to take into account in human detection [6].

2.2.4 Scale Invariant Feature Transform (SIFT)

The SIFT descriptor extracts relevant points in a training image, which can then be used to find that image in a scene that can contain many other objects [23]. In order to achieve a reliable recognition, which should work even if the object to detect suffers noise, scale, orientation or illumination changes, the features are usually extracted from high contrast regions, such as edges, either between the object and the background or within the object.

Despite providing good reference points, this descriptor is more useful when the object to detect either matches previously known objects, or is very similar to them, as it retains texture and small detail information (fig. 2.2c). Therefore, its applications include panorama stitching software, satellite image processing [24] and robot localization [25].



(a) Source image (b) CENTRIST (c) SIFT

Figure 2.2: Codebook visualization from CENTRIST and SIFT [2]

2.2.5 Covariance matrix

Covariance descriptors extract feature matrices which can map one of many image characteristics, such as color, gradients, or intensity [26]. These covariant matrices describe the relation between the points inside them, and are usually enough to match that extracted region even with geometrical transformations applied.

The good level of invariance to rotation and distortion is precisely one of the strengths of this descriptor [27], making it useful in situations where the position of the detection targets is

unpredictable. This is not the case in this dissertation work, as the subjects to be detected are expected to be in a standing position.

2.3 Human detection frameworks

The field of computer vision technologies is currently the subject of many academic research projects, yet improvements still come in small amounts, as most research activity builds up on one of the previously available frameworks. Therefore, this section covers the frameworks that are currently regarded as the best in the field, and highlights some of the recent improvements that have been made.

Histogram based detectors usually implement the HOG descriptor as their main feature extractor, and can combine with other descriptors and classifiers to improve the detection results. Within this category, the first and still one of the most used implementations of the HOG descriptor is presented in the paper *Histograms of Oriented Gradients for Human Detection* [3]. This particular implementation combines a histogram-based descriptor with a spatial histogram normalization method (SIFT [28]).

The results provided in the paper were obtained from tests on two different datasets:

- the MIT pedestrian database, containing only front or back views and a limited range of poses;
- the Inria database, developed by the paper authors and providing a bigger challenge for the descriptor, by containing images of people captured while standing or moving in a natural way, with a height of at least 100 pixels, in a wide range of backgrounds, including crowds. However, in this dataset, the positive samples were cropped around the subjects to be detected, to reduce the original resolution, which also has the effect of reducing the false positive detections.

For the MIT dataset, the descriptor performed near-perfectly, with a miss rate of less than 1% at 10^{-4} false positives per window (FPPW). It was due to these results that the Inria dataset was developed. Its miss rate, for the same FPPW rate, is of 10%. While these results are worse than for the MIT dataset, they still provide a big improvement over other descriptors. All these results can be seen in figure 2.3.

The improvements HOG brought into the human detection field and its ease of implementation have led to other detectors using it as either a starting point, or a step in the chain of image processing. However, this could bring with it some drawbacks, as the weaknesses of the Histogram of Oriented Gradients approach will affect other detection solutions if they are not addressed.

Another histogram-based solution is the HOG-LBP [21]. This detector combines the framework described in [3] with a Local Binary Pattern, in a detector that also handles partial occlusion by taking advantage of the LBP high discriminative power, along with the HOG edge and local shape information capture.

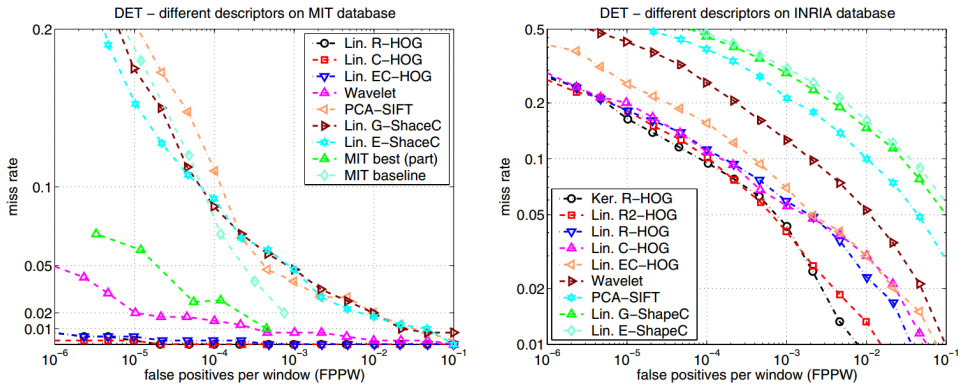


Figure 2.3: Detection performance of the HOG implementation from Dalal and Triggs [3]

After computing the Histogram of Oriented Gradients and the Local Binary Pattern values, these are combined in an augmented feature vector. It is on this vector that the sliding window acts. By feeding the sliding window results to a Support Vector Machine (SVM) [29], each block can be scored. If the SVM scores the block with an ambiguous classification, an image segmentation algorithm is run, which segments the possible occlusion regions.

The HOG-LBP method achieved a detection rate of 91.3% at 10^{-6} FPPW and 94.7% at 10^{-5} FPPW using the Inria dataset. This compares with the HOG rate of 90% at 10^{-4} FPPW.

Despite increasing the detector complexity, this method proved effective at detecting partially occluded subjects in the tested datasets. As a negative point, it should be noted that this approach is much slower than HOG on its own, which is already relatively slow.

There are also attempts to enhance the HOG feature extraction method by complementing it with better segmentation methods, employing, for instance, soft segmentation using color information. In [15], the CHOG approach (Color-HOG) is proposed. This method aims to segment the pixels in a given region into foreground and background, by taking advantage of RGB data. For all HOG blocks, it runs a semi-local segmentation method that takes a set of reference pairs of image points and defines them as background or foreground, as estimated from their neighboring pixels.

The results show a small improvement over the traditional HOG implementation. For the same 10^{-4} FPPW, the miss rate drops from 10% to 6.5%.

Another feature combination method concatenates the CT with the HOG descriptors [4]. While the combination of two feature extraction methods usually decreases performance, this approach also takes into account the redundancy in the histograms of adjacent windows [30], computing only a few histogram bins. This concept is applied to the HOG features, as well as in the CT computation.

The miss rate obtained with this method is lower than with both HOG and CHOG, at 4.5% for a 10^{-4} FPPW rate (fig. 2.4), while processing images faster than the other methods [4]. These

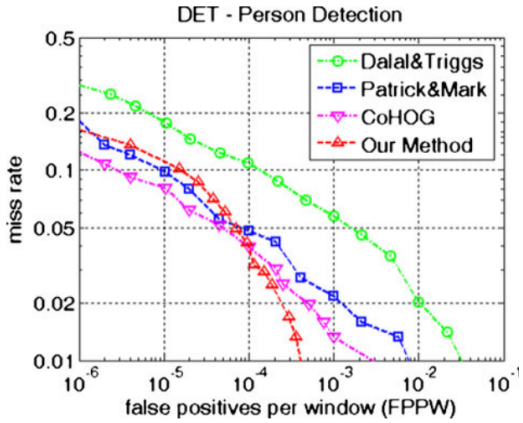


Figure 2.4: Detection performance of the HOG-CT implementation from Ding et al. [4]

values encourage the use of multiple, complementary feature extraction methods in the same solution, or a combination of their outputs, which could enhance the results obtained.

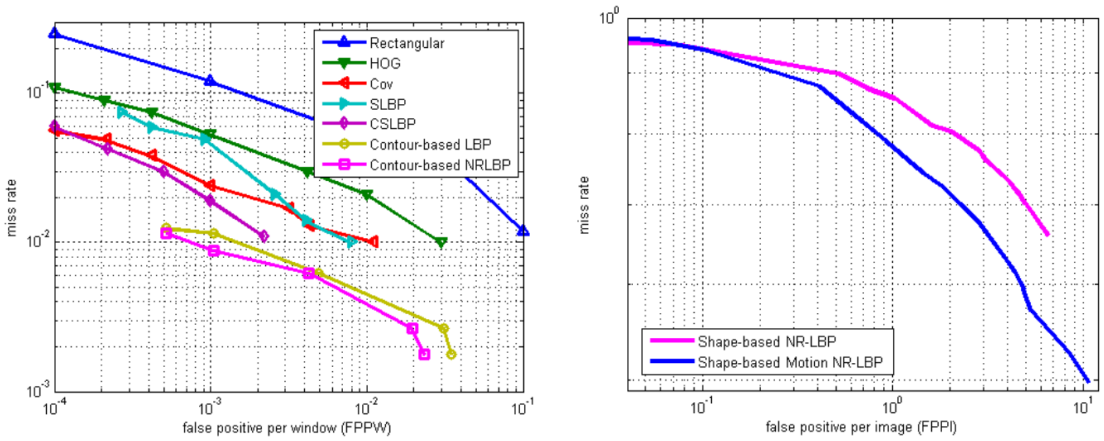
When the focus of the detection are video sequences, the existing implementations tend to take advantage of movement information. Such is the case with [5]. This method uses a variant of LBP, called Non-Redundant Local Binary Pattern. This descriptor is more discriminative and robust to background, foreground, and illumination changes than the original LBP, in which it is based. The motion information is used by taking movement shape templates and matching them with the input images, using a detection window. Then, the difference image between two consecutive frames is computed, and for each point in the matching template, the corresponding point in the difference image is found. Afterwards, a set of feature vectors is created, corresponding to the object movement.

According to the results provided, an improvement over HOG was achieved in both video detection and on the Inria dataset. At a 10^{-3} FPPW rate, this method achieved a miss rate of about 1%, while HOG misses 5% of detections (fig. 2.5a). As for video, the advantages that arise from using motion information can be seen in figure 2.5b.

While this method provides a good approach for video processing, by taking advantage of the inherent motion information, the quality of this information is also a relevant factor. If the frame rate of the input video sequences is too low, motion information cannot be used positively, making the detection a problem that needs to be solved frame by frame.

Contour-based detectors take on the notion that human shapes are usually distinguishable from the background they are in by determining their contour. By computing all the shapes in an image, the descriptor makes it easier for the detector to then find the subjects in the scene.

As the CENTRIST descriptor allows for a good evaluation of scene elements, despite not directly implementing object detection, the features it extracts can be used by a human detection



(a) Inria dataset (b) ETH video dataset

Figure 2.5: Detection performance of the video oriented implementation from Nguyen et al. [5]

classifier. For this, other implementations were developed, such as the C^4 detector, by Wu et al. [6], which is a CENTRIST-based detector that focuses on contour cues. C^4 works by first creating the Sobel gradients of the image, then computing the Census Transform values and creating a single integral image. This image is then resized and the brute-force scan is performed.

The major performance advantage comes from only using one integral image, and from the fact that CENTRIST does not require normalization, unlike HOG.

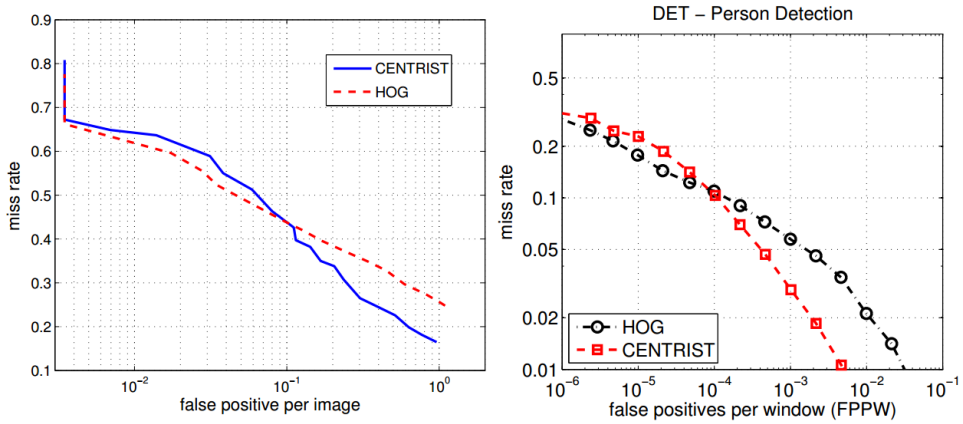


Figure 2.6: Comparison between C^4 and HOG on the INRIA dataset (from [6])

This detector achieves better results than HOG at higher detection rates, while being slightly worse when very low FPPW values are targeted. It is also more resistant to cluttered backgrounds than HOG, as the Sobel image smooths the information from high frequency local textures, making it a good match for a dual algorithm detection solution. A comparison with HOG can be seen on image 2.6. When comparing detection speed, C^4 can process a 640 by 480 video at 20 frames per second (FPS) using 1 core of a dual core 2.8GHz CPU. The nearest comparable solution ran

at 10 FPS, while also using parallel processing on a GPU.

Other detection frameworks implement different ways of processing the images and doing the detection. For instance, the *Viola–Jones object detection framework* [31] describes a method of detecting objects in a scene. Its mostly used implementation focuses on face detection.

This procedure works by classifying images based on simple features, which makes for a much faster processing, allowing the focus to be put on the quality of the results. By also implementing a cascade of classifiers, this method can achieve good detection rates (>85%) while providing low false positive rates (<10^{-5}), as the detection results are improved along the processing pipeline. A target detection rate of 0.9 (90%) can be obtained using a 10 stage classifier. Each stage needs to have a detection rate of 0.99, but can also have a relatively large margin of error, as false positive detections will be sequentially eliminated in the following stages. As a result, with this implementation, each stage can have a false positive rate of 0.3 FPPW. The end rates would be:

- Detection rate: $0.99^{10} = 0.904 = 90.4\%$
- False positive rate: $0.30^{10} = 5.9 \times 10^{-6} = 5.9 \times 10^{-4}$ FPPW

As for its speed, on a 700 Mhz Pentium III processor, using 384 by 288 images, each one took 0.067 seconds to process, which translates to 15 FPS.

The Viola-Jones framework provides good results at a fast speed, which can be used by a complete human detection solution, as by detecting the face first it could then more easily detect the rest of the person. In this dissertation context, it could be useful as either a first step processing or as a false positive filtering method, if the image quality is good enough to allow for a good face detection implementation.

This type of approach has also been used in pedestrian detection [32], by taking advantage of motion information. The results from that implementation show different ranges of improvement. In one test sequence, the difference to a static detector is minimal. However, in a different sequence, the static detector achieved a detection rate of 56%, while the dynamic detector got a 90% detection rate, both at a $2x10^{-5}$ FPPW rate. The test sequences used in these results show well illuminated areas and present a very low amount of compression artifacts.

Chapter 3

Experimentation Framework

Before the solution development could start, a series of tasks needed to be accomplished. Those tasks, which are presented in this chapter, include the video sequence capturing, the generation of the ground truth for those sequences and the metrics to be used for evaluating the final results.

3.1 Sequences

Although the main objective of this dissertation is human detection, the sequences recorded also took into account future development projects, such as targeting and the analysis of consumer behavior models.

As legal reasons prevent the use of recordings without the consent of the people present in the video, and in order to emulate the most relevant behaviors and represent a wide range of possible scenarios in the shortest amount of time, a script that took into account the way costumers behave while shopping in a retail store was used, based on [33] and [34]. This script contained 10 scenes in which the actors would follow certain actions, which would encompass a wide range of consumer behaviors and characteristics, like walking into certain directions and areas of the store, touching products, gestures and conversation. The script was adapted in a way that made feasible its execution, regarding both the time frame available for capturing the sequences, and the number of participants. A copy of the final script is included in appendix A.

After obtaining the sequences from the video recording equipment, their quality was evaluated. The video presents a resolution of 640x480, at a frame rate of 1 FPS. From this information, the frame rate is already revealed to be a barrier for certain applications, such as tracking and the use of temporal information. The VGA resolution seems to be appropriate, yet the perceived image quality is much lower that what the resolution implies.

The sequences obtained contain a large amount of encoding artifacts, making regions of the video that should be static appear to have movement from frame to frame. The details were lost even further in dark areas, as darker regions were encoded as having only one color, resulting in a more difficult distinction between background and foreground (fig. 3.1a). Despite the overall dark environment, all ceiling lamps and outside lights presented in the store affected a large area

around them when the camera was pointed anywhere in their vicinity (fig. 3.1b). As an effect, everything in those areas became washed out in the video, again contributing to a significant loss of detail and contrast of the image. Lastly, the presence of macroblocks can be easily seen in the whole captured area, difficulting the work of any contour detectors employed (fig. 3.1c).

Seven video sequences were obtained, and the most relevant parts from each were selected, making sure only the scenes from the script would be part of the final dataset. After the selection process, each remaining sequence contained between 500 and 800 images.

3.2 Ground truth

In order to evaluate the solution developed in this dissertation, and to allow the dataset to be used in other projects, the sequences recorded were manually annotated.

For the ground truth generation, a program that generates a Computer Vision Markup Language (CVML) [35] formated output, based on the selections made, was used. An example of the output created and the bounding boxes from that sample can be seen in figure 3.2.

The image quality was a challenge also during the ground truth generation. Due to the dark environment in the store and the bad video encoding, dark clothes tend to lose their definition and are encoded as a large semi-uniform region, without enough variation from the background, thus making it almost impossible to detect their contours. Other times, people farther away from the camera were encoded with too many artifacts and could not be recognized even by a human, and so were not marked in the ground truth. These exclusions will, however, slightly increase the number of false positives, as the people excluded can very sporadically become visible enough for the detector to register them in a frame.

3.3 Metrics and evaluation

The evaluation of the algorithm's results is an important step in the process of judging its effectiveness at detecting the subjects in the sequences.

For an accurate assessment of the output results, a set of scripts implementing frame based metrics, proposed by [36] and which have also been used in papers from other authors [37, 38], was used. The metrics described in [36] cover such performance elements as detection rate, accuracy, false positive and false negative rate, and predictive value. However, since the main objective of this dissertation is the detection phase, when comparing results from different solutions the focus will be placed on the False Alarm Rate, Detection Rate and Accuracy metrics, and not on the tracking measurements.

In the metrics used, detections are taken on a frame basis. This means, for example, that a true positive result occurs if both ground truth and detection results agree that an object exists in a particular frame, and the bounding box of at least one of those objects coincides between ground truth and detections. For that match to occur, the center of the bounding box of an object must fall within the limits of the bounding box of the other object. As for what the metrics refer to, False

Alarm Rate, or FAR, is the percentage of total detections that are false positives (Eq. 3.1). The Detection Rate, or DR, represents the percentage of objects correctly detected (Eq. 3.2). Lastly, Accuracy, or Ac, accounts for the percentage of frames that were correctly processed (Eq. 3.3).

$$FAR = \frac{FP}{TP+FP} \tag{3.1}$$

$$DR = \frac{TP}{TP+FN} \tag{3.2}$$

$$Ac = \frac{TP+TN}{TF} \tag{3.3}$$

During the solution development, using only the metrics to evaluate the results did not provide enough feedback to make a good assessment of the steps that were being taken. Since the metrics can only reveal the raw detection rates, a direct comparison between sequences processed with different parameters was needed. This way, a proper analysis of the impact that each change in the image processing algorithms was having could be established, and both positive and negative effects could be observed. This approach does not serve in any way as an overall assessment of the solution performance, instead being a subjective evaluation of the development progress. As a good and viable software for comparing two sequences of images side by side was not found, a solution was developed, as seen on figure 3.3.

(a) Loss of dark details

(b) Details affected by electrical lights

(c) Detail of the macroblocks, zoom factor 2x

Figure 3.1: Examples of the challenges presented by the quality of the video sequences

```
<?xml version="1.0" encoding="UTF-8"?>
<dataset name="07_Camera7_00001.jpg">
(...)
 <frame number="12">
     <objectlist>
         <object id="1">
             <orientation>90</orientation>
             <box xc="489" yc="357" w="59" h="180"/>
         </object>
         <object id="2">
             <orientation>90</orientation>
             <box xc="509" yc="302" w="86" h="173"/>
         </object>
     </objectlist>
     <grouplist/>
 </frame>
(...)
</dataset>
```

Figure 3.2: Sample of the CVML output and corresponding bounding boxes

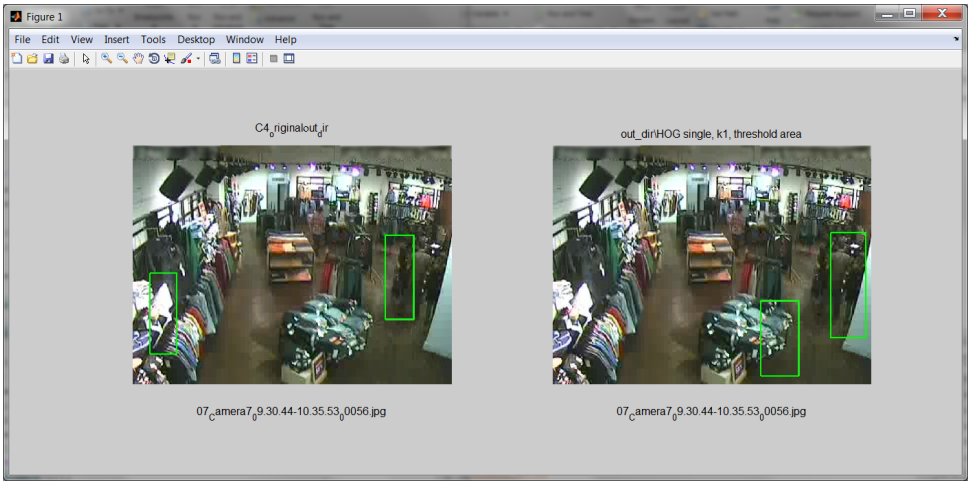


Figure 3.3: Side-by-side image comparison

Chapter 4

Detection Solution

The development of the human detection solution proposed in this dissertation started with a pre-processing stage, where various image and video processing methods were tested and applied.

All tests were run on the same hardware, on an Intel Core i7-4700MQ [39] at 2.3Ghz with turbo boost disabled, to minimize speed fluctuations.

4.1 Algorithm selection

In order to assess the performance of the filters and processing methods being considered, detection algorithms would need to be tested and evaluated, so the best ones could be used in the final implementation.

Based on the concepts and approaches described in section 2.3, some available algorithms were collected for testing. In order to be useful for this dissertation, they had to output a file with some type of coordinates from each detection that could then be processed or even converted to CVML. Algorithms that implemented HOG, HOG-LBP and C^4 were obtained and tested. Since the low quality of the images revealed to be to harsh even when creating the manual ground truth, approaches based on facial detection were discarded from the start.

The first tests were conducted with an HOG implementation. A wrapping was performed so the program would also output the images with a bounding box, marking the location of each detection.

The results were promising, with many positive detections made, but with an even higher number of false positives. However, this huge amount of false positive results can only be fully understood when visualizing the detection output, as shown in figure 4.1. These results are not accurately reflected in the metrics, as those calculations are made taking into account the results each frame as a whole, and not each individual false positive detection. Since the false positives could be drastically reduced by the background subtraction filters, and considering the good true positive detection results, the HOG detection algorithm was still accepted for the final implementation.

Figure 4.1: HOG detection results

An HOG-LBP implementation [21] was also tested. For this program the source code was not available, but its output still produced the detection windows over the images and a text file with their respective coordinates.

This method revealed to be extremely slow, taking about 10 seconds to process each 640x480 frame. Yet, the detections were scarce. From a sequence of 400 frames with at least a person in nearly every image, only 30 detections were made (fig. 4.2), and only once the algorithm made two detections in the same frame. The few detections made were generally accurate, but attending to the low detection rate and the unavailability of the source code, this implementation was not used in the final solution.

Figure 4.2: Two of the 30 HOG-LBP detections on a 400 frame sequence

Tests were also conducted with the C^4 implementation, obtained from [6]. Some changes were made to the code provided, in order to obtain a proper output visualization.

This algorithm produced balanced results, with less positive detections than HOG, but also far fewer false positives. This method was also the fastest, with about 6 images processed per second. Some of its detection results can be seen in figure 4.3.

Figure 4.3: C^4 detection results

After the tests were run, the HOG and C^4 algorithms were integrated in the solution being developed.

4.2 Image processing methods

4.2.1 Super-resolution (SR)

Super-resolution is an image processing technique used mainly to enhance details in low quality images. It works by taking advantage of small differences between images in order to create a new image with enhanced details and, usually, higher, upscaled resolution.

The differences between images should derive from sub-pixel displacement, such as small variations in lighting, capturing angle (small shaking of the hands of the photographer), motion blur and noise [40]. However, these variations are not very prominent in fixed position cameras on an interior location, which ended up compromising the expected results negatively.

Another factor that affected the SR results was the movement of the people in the video, as their change in position between sequential frames is usually too big to give a positive contribution to this process, and the algorithm either does not enhance the subject in a substantially better way than a sharpening mask, or creates a ghosting effect. This ghosting effect becomes even more predominant with the increase of the number of samples used, as the SR methods need the largest sensible amount of source images in order to provide the best results.

As can be observed in figures 4.4a and 4.4b, the use of the minimum amount of samples required by the super-resolution algorithm creates a strong ghosting effect on the moving subjects, located on the left of the frame. Using only two sequential samples to compute the SR image reduced the ghosting effect, but did not provided enough improvements, as two samples are not sufficient to obtain relevant results. When the same sample image is used twice, to prevent the ghosting effect (figure 4.4c), the image details are not enhanced, and a slight blur can even be seen in the full scale image. The sharpening method, presented in 4.4d and implemented via an unsharp mask on Adobe Photoshop, revealed to increase some details without creating ghosting effects, but

(a) SR using 2 samples

(b) SR using 3 samples

(c) SR using the same sample twice

(d) Unsharp mask, using Adobe Photoshop

Figure 4.4: Comparison between 3 super-resolution images and a sharpening filter

was also not used, as it made the compression artifacts even more noticeable, which would make the process of distinguishing the background from the foreground subjects more difficult for the detection algorithms.

The ineffectiveness of the super-resolution algorithms also comes, in part, from the fact that most of them are tested by downscaling one high resolution image using random warping, blurring, and noise filters and then re-upscaling and comparing it with the original image [40] [41]. The problem with this validation method is that it makes this approach only applicable on photos and not on regular video, where the subjects change position between frames. For it to be useful for video, the subjects would either have to be almost static, or the recording would otherwise need to be done at a very high frame rate.

In the end, the super-resolution method was disregarded and not used in any stage of the solution developed. It could only be applied in situations where the subjects were practically static between frames and this would not be the case in a retail store. If it were the case where a super-resolution algorithm could be applied, the resulting improvements would still need to justify the processing time, which they do not seem to do for this kind of sequences. Due to low quality

encoding quality, a sharpening mask was also not applied, as it made the encoding artifacts even more visible, instead of improving the overall image quality.

4.2.2 Dilation

This mathematical operation works by computing the maximum pixel value that results from overlapping an image with a matrix, for every image pixel. The value located in the center of that matrix is replace by the maximum computed from the other matrix positions, resulting in an image with expanded brighter areas, in relation to the source image [42].

The dilation can be configured in different ways. For instance, to change the intensity of the dilation, as seen on image 4.5, the kernel value that is used as a parameter to the dilation function should be changed. The dilation can also take different shapes, such as rectangle, cross, or ellipse. For these sequences, the rectangle dilation was chosen, as it caused the less distortion.

(a) Source image

(b) Dilation with kernel size 1

(c) Dilation with kernel size 2

Figure 4.5: Comparison of different dilation factors

The application of this filter helped improve the detection results, as the images became slightly lighter, and people in the sequences were more easily distinguishable from the background. In one of the test sequences, the usage of this filter improved the detection rate, from

63.7% to 68.2%, and the false alarm rate, which went from 24.9% to 18.5%.

4.3 Temporal information

The fact that the dataset in use was not a group of unrelated still images, but video sequences, allowed for the possibility of using temporal information to help improve the final detection results.

The analysis of a sequence of images over time can provide information that helps the estimation of where a detection could occur next, based on previous detections and current movement direction. A separate processing method could also keep track of low probability detections (if the human detector in use assigns probabilities to each detection) and confirm or discard them if they made sense on a temporal basis. For instance, if a detection occurs in an area around which no other detection is present in previous or future frames, that detection has a higher chance of being a false positive. By that same logic, if a detection does not exist in a frame, but is present in the previous and next frames, an estimation could be made to correct the false negative. However, the implementation of this method should take into account that the missed detection could be due to a subject occlusion, and not to an algorithm flaw.

The first approach considered was Optical flow [43]. This method creates motion vectors for the sequences, allowing for an estimation of the translation that each detected subject suffers between frames. As with other state of the art approaches, the results presented in [43] were tested in well illuminated areas, with good frame rates and virtually no objects that could overlap with the subjects in the video.

After tests were conducted with a real-time post-processing software that generated the motion vectors on-the-fly [44], the results can be seen on figure 4.6. For many images the motion vectors were not even found. When they were determined, there were too few of them, and usually inaccurate, either pointing in the wrong direction, or with a length that did not match the movement. While testing on other footage, such as sequences from the 2008 edition of TRECVID [7], with not only better overall image quality but also a frame rate of 25 FPS, the motion vectors appeared in much higher quantity and with better accuracy.

As such, due to the previously mentioned low quality of the video sequences targeted by this dissertation, this method could not the applied in the detection solution. The difference in the position of each subject in consecutive frames is too big for this approach, as the motion estimation cannot guess with high accuracy where the blocks of pixels from the previous framed moved to. When expanding the searching area, more groups can be considered as candidates, but then too many false positive matches would be found, making the whole process inaccurate.

The other approach was track-before-detect, from the paper *Multi-camera track-before-detect* [45]. The principle behind it is that a signal from an image may be too weak to provide a positive detection with just one sample, hence by tracking and integrating it over time a more accurate assessment can be made. Unfortunately, the same drawbacks as with Optical flow revealed themselves: for an effective use, this method would need a good frame rate and a better image quality.

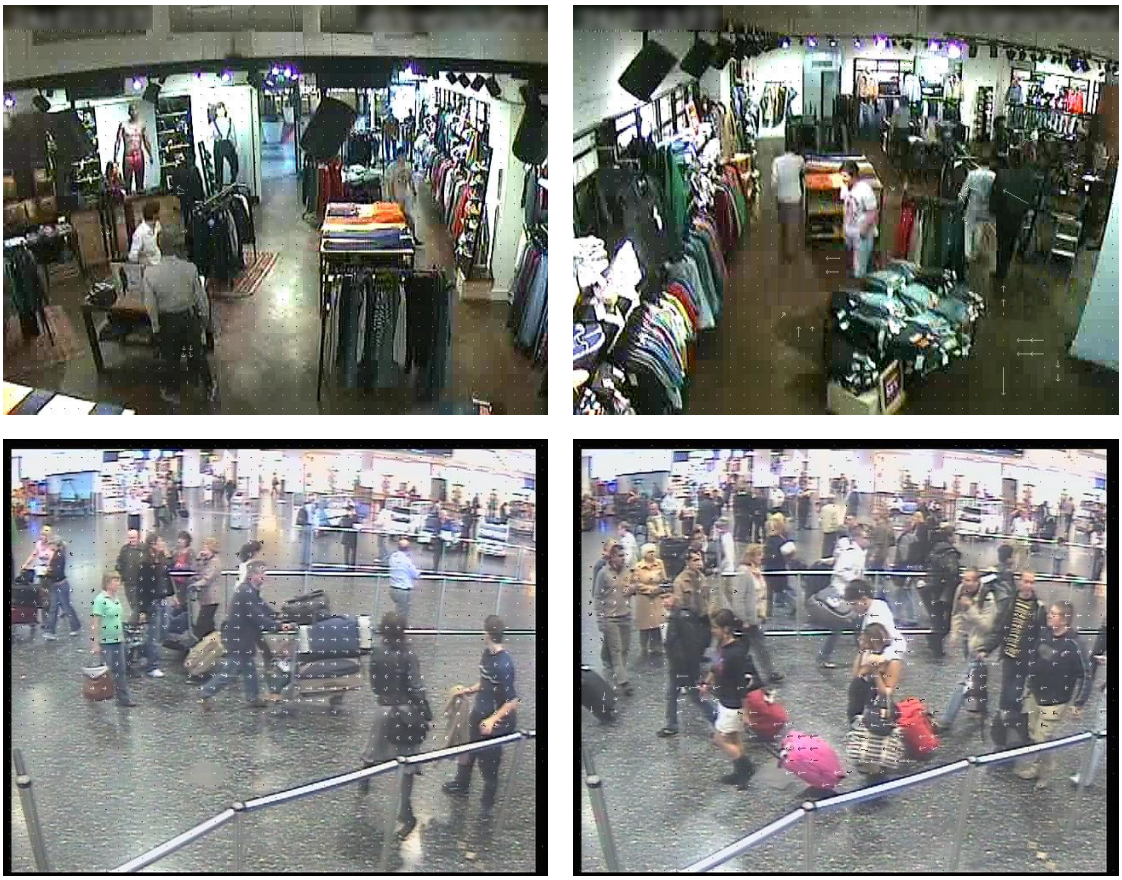


Figure 4.6: Motion vectors: Dissertation sequences vs. higher quality video [7]

Both these approaches seem to require the detection on which they would be applied to provide good baseline results by itself, with a high detection rate and low FPPW rate, so those results could then be taken a step further. That was also not the case with these sequences, as will be described in chapter 5.

4.4 Background removal

After some tests with the algorithms selected, HOG and C^4, revealed very high FPPW rates, a decision was made to reduce those values by implementing a background removal algorithm in the preprocessing stage. The two methods chosen, MOG and median, described in chapter 2, were tested on the available sequences.

4.4.1 Foreground masking

The first tests were conducted using the MOG implementation from the OpenCV documentation site [1], and the results were not great, again due to the low quality of the sequences. Some background elements still made it through to the final image and, more importantly, a big part of

the foreground elements was cut off, as seen on figure 4.7b. The resulting mask could have been processed further, but it would be very difficult to accurately fill in the gaps in the mask while also keeping the mask close to the subjects, to avoid the false positive detections that this method was supposed to help reduce. Therefore, an implementation of the median method was developed, allowing for a deeper level of customization that would target this kind of sequences, as they were of a very different quality than the ones usually used while developing and testing new algorithms.

The median algorithm developed works by processing a sequence of images, selecting a predetermined number of them with a certain interval between each selected image. These images were then processed, using a dilation filter, and the median was computed. This process was accomplished by adding the values of the same corresponding position from each image and determining its median value, using a partial sorting algorithm. Repeating the process for every position in the images resulted in the median image, containing only the background of the scene. A Gaussian blur was then applied to the resulting median image, making it less sharp, which gave a little margin of error to each position. As a result, each pixel value was influenced by its neighbors, as also occurred in the source images due to the bad compression algorithms which made the value for each position slightly change between consecutive frames, even with no changes in light, or any movement around that pixel.

To then obtain the foreground mask for each frame from the sequence, each image went through another step before being processed by the human detection algorithms. In this process, a dilation filter with the same parameters as the one used in the median algorithm was applied to each image, and then the median image was subtracted from the image being processed. When the result of a subtraction was lower than a determined threshold, it meant those pixels were very similar, and thus part of the background, being that position marked as black in the masking image, while the remaining pixels were kept white. This resulted in the foreground mask, as seen on figure 4.7c, the same type of output that was created by the MOG algorithm, but with results closer to the ones desired.

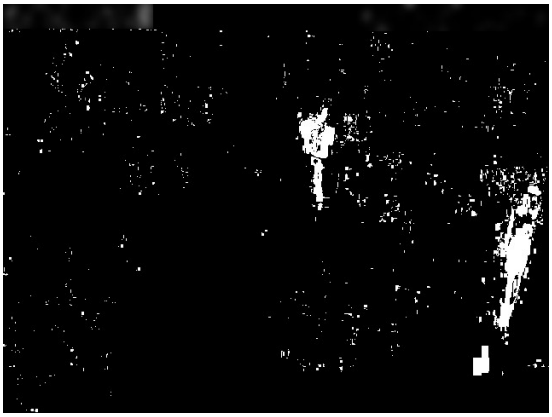
For an appropriate configuration of this algorithm, the number of samples to take, the interval between them and the subtraction threshold were obtained by experimenting different parameters and selecting the ones that provided the best results.

The key difference between the two output masks seen on figure 4.7 is the higher detail of the foreground elements. In the median mask, the shape of the people presented in the scene is recognizable even on a black and white image, which will enable a much more accurate bounding box around the subject, while in the MOG mask there is no way of knowing how the subjects are oriented in the scene, or if the mask contains the lower or higher part of their body.

In the MOG mask, the inclusion of parts of the background that are the same size, and sometimes even bigger than the foreground elements would also increase the area of the image that would be processed by the detectors, diminishing the advantages of using a background subtraction during the preprocessing stage. While in the median filter a higher number of background elements are present, they are usually of a size much smaller than the foreground elements, making it easier to filter them out before the background subtraction is made.

(a) Source image

(b) MOG mask



(c) Median mask

Figure 4.7: Comparison of two foreground detection methods, MOG and median

4.4.2 Background subtraction

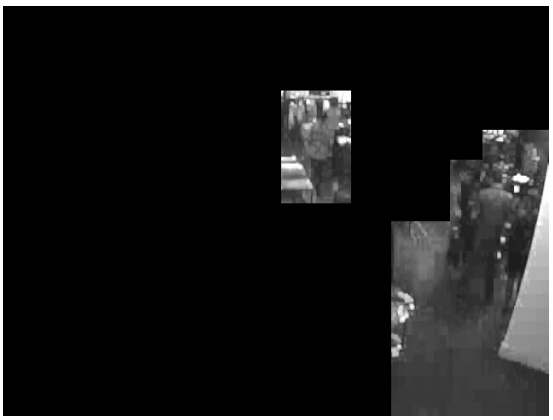
For the background subtraction, an approach based on contour curves was used. This way, the smaller elements from the mask generated by the median algorithm could be discarded. Working with contour curves also enabled the use of some auxiliary functions that made possible the calculation of the center of mass of each foreground zone, making the process of creating a bounding box around that zone easier and more accurate.

First, all the contours from the foreground mask were detected, which resulted in a series of contours of a wide range of sizes. Then, by processing only the ones with an area above a threshold that could indicate the presence of a person (also determined by experimental results), most of the previously mentioned background elements that made it through to the mask were ignored. As the remaining contours were located on the regions where people could be, a bounding box around each of them would need to be created.

In order to determine an accurate bounding box to create around each contour, a box that also accounted for the variation of size with perspective, the position of the center of gravity of each contour was determined. In this step, the moments for each contour were computed, and the

respective mass center was calculated [46].

(a) Source image



(b) Foreground elements

(c) Foreground elements isolated

Figure 4.8: Results from the background subtraction method

As for the horizontal and vertical margins around each mass center, they were based on the coordinates of that center, so as to include a perspective factor. When the perspective from a typical surveillance camera is taken into account, subjects near the top of the image are farther away from said camera, and thus appear smaller. The resulting implication is that centers of mass with a lower vertical coordinate value should have a smaller bounding box around them, as long as the coordinate system in use is centered on the top left and oriented positively to the bottom right, as the one used in the OpenCV Matrix Data Type [47] (fig. 4.8a).

In the end, a bounding box was calculated from each contour, and the corresponding area was obtained from the source image and copied to a final image (fig. 4.8b). This image contained only the foreground areas that the human detecting algorithms would work on, helping eliminate false positives, as was desired. Another set of small images was also created, by separating each area from this final image into a segment (fig. 4.8c). These segments could also be used by the detection

algorithms instead of the bigger single image, as they provided only slightly worse results but were much faster to process, due to their smaller size.

4.4.3 Scene masking

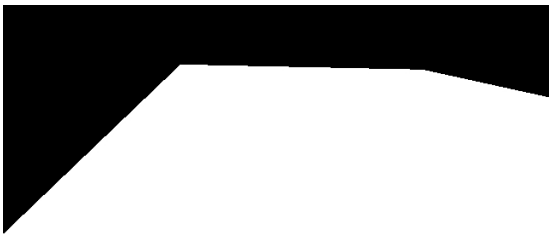
The automated process of background subtraction can also be complemented by a scene mask (fig. 4.9). This mask should be created manually for each camera, and will be added to the foreground mask generated by the software.

The objective of the scene mask is to permanently eliminate parts of the scene where no person will ever appear, according to the camera point of view and the architecture of the space being filmed. In these sequences, for instance, the ceilings were masked in black, as were the higher part of the walls. It should be noted that mannequins and tables should not be marked as part of the background even if they are not moved, as detections can still be made when a person passes in front of them. These mannequins presented in the scene can also affect the detection results, as their shape resembles that of a human, and can therefore be a source of false positive detections.

(a) Source image from camera 2

(b) Scene mask for camera 2

(c) Source image from camera 7



(d) Scene mask for camera 7

Figure 4.9: Background masking from the point of view of two cameras

Chapter 5

Implementation and Results

The workflow of the solution developed and the results obtained are presented in this chapter, along with their analysis.

5.1 Integration and workflow

The developed program combines the preprocessing, the detection algorithms, and post-processing stages. Its workflow starts by taking the list of frames to be processed. Before the first run, the background image is calculated by taking 5 samples from the list, starting with the first image and taking the other 4 with a 3 frame interval between each of them. From this point on, the background image will be updated each 8 frames. Near the end, if an update is required and there are not enough samples left, the last image in the sequence will be used, and the other samples are counted backwards from there. All configuration values were determined by running the solution multiple times with different parameters and finding a combination of values that maximized the quality of the detection results.

After the background is computed, the first image can be processed by the segmentation algorithm. As described in section 4.4, the image is masked to remove its background elements, leaving only the foreground objects surrounded by a bounding box. The image that derives from this process is then sent to both detections algorithms, HOG and C^4. These will run on two separate threads, so the overall processing time is only the slowest of the two, instead of being the sum of the time each method needs. These algorithms will process the image and perform the detections, which are saved independently.

Before the results are saved to a file, the output that each algorithm returns is processed and combined in a different step. First, any detection whose bounding box is located inside the limits of another detection is removed, with only the outside result being kept. Then, each detection from one algorithm is checked against all detections from the other, two results at a time. If all the limits of their bounding boxes differ only by less than 10%, the result from the HOG algorithm is discarded, as this has been found to be the least accurate in bounding box placement. An

illustration of this process can be seen in figure 5.1. The final results are then written to the output file, using the same CVML formatting style as in the ground truth generation.

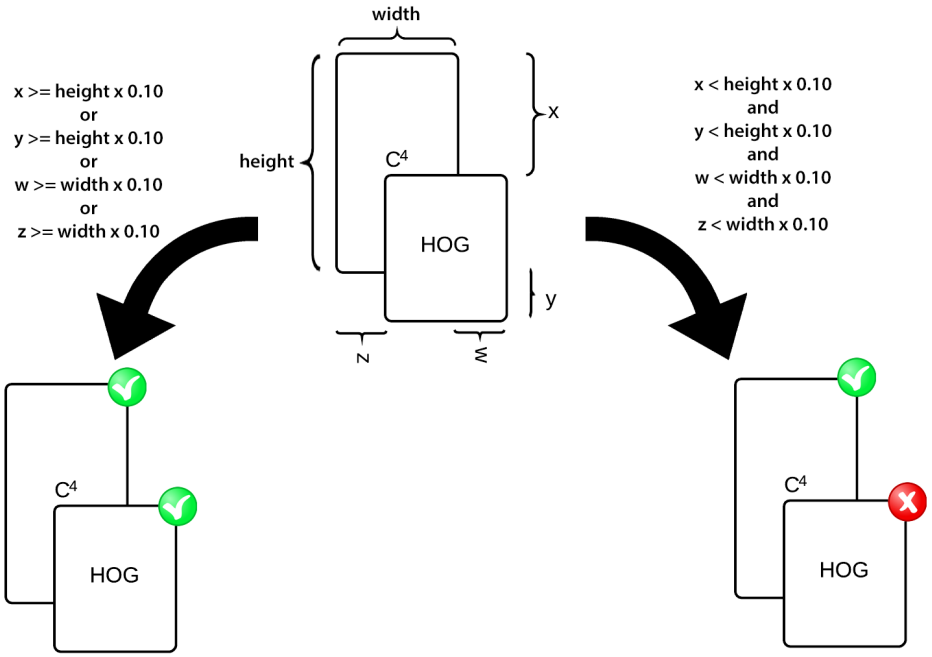


Figure 5.1: Results combination

This process continues until all images are processed. A flowchart of this workflow can be seen in figure 5.3, while the application in work can be seen in figure 5.2, with the respective output.

5.2 Results

When the development was concluded, each sequence was processed by the defined solution and by the basic HOG and C^4 implementations. The detection results were compared to the ground truth and the metrics were calculated.

The solution developed improved the results on all metrics. The False Alarm Rate fell drastically, as the chance of the detectors finding false positives was reduced by the removal of most background elements from the images. This fact, combined with the image processing to improve clarity, allowed for improvements in the Detection Rate. By not filling up the image being processed with wrong detections, the algorithms could actually make the positive detections that were expected. Both these improvements reflected on the Accuracy metric, with more frames being processed correctly than before.

The False Alarm Rate fell to less than half of the one achieved by the C^4 algorithm (table 5.1). The difference to HOG is not as drastic in the metrics, being of almost 15 percent points on average, which does not reflect the differences noticed in the image sequences, due to the metrics

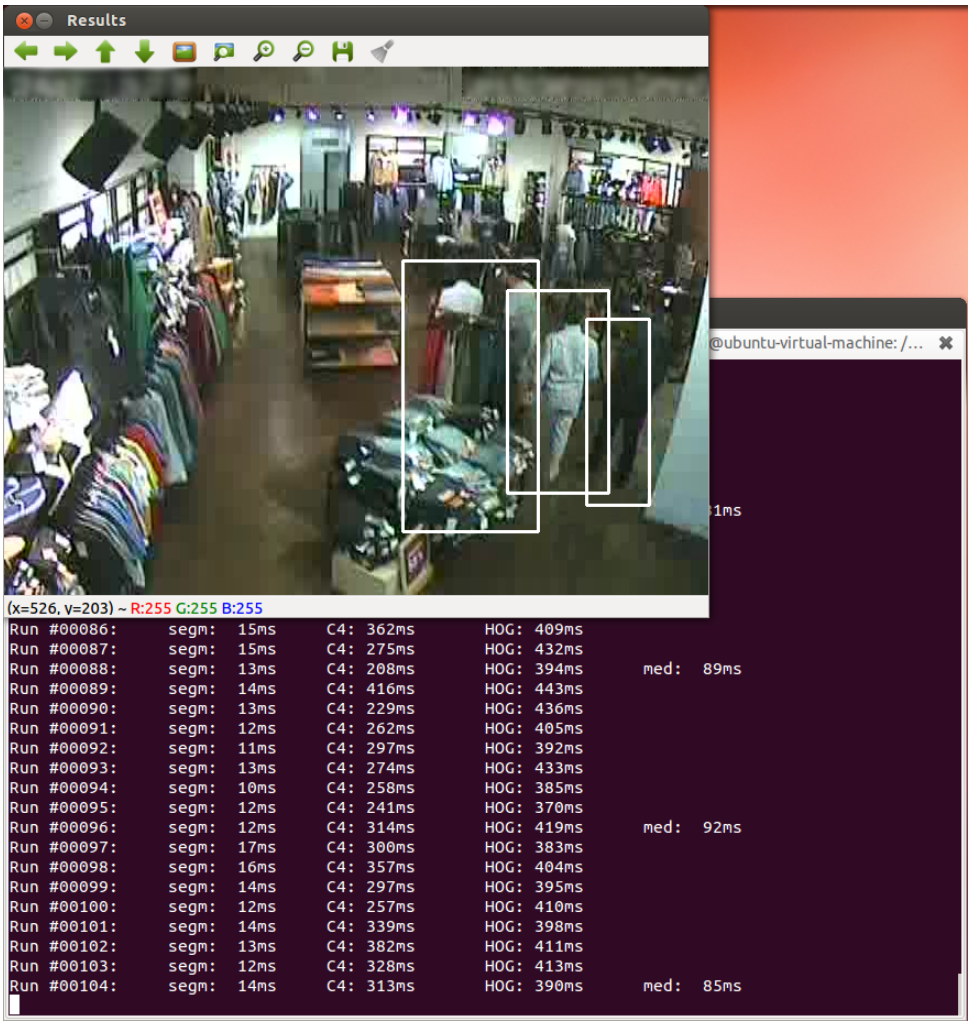


Figure 5.2: Detection in progress and output information

used. Therefore, despite the HOG algorithm having much more total false positive detections than any of the others, these usually occurred grouped in a frame, which would count as one bad detection in the metrics, while the following frame could have less false positives, skewing the final results. An example of this occurrence can be seen on figure 5.4, where the results from HOG show a much larger amount of false positives, while also providing some true positive detections. C^4 only made one detection, and the solution developed made 3 of the 5 detections possible, while also detecting a mannequin that was not removed by the background subtraction process due to the presence of a foreground element nearby, in this case the person in a white shirt. As such, the bounding box from that foreground element must have also included the mannequin, resulting in the false positive.

Both the accuracy (table 5.3) and detection rates (table 5.2) were also improved by nearly 30 percent points when compared to C^4, and almost 10 percent points relative to HOG, which also had a much larger false positive rate.

False Alarm Rate				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	44.48%	25.79%	43.98%	38.08%
C^4	68.08%	52.71%	45.16%	55.32%
Solution presented, HOG only	32.34%	25.73%	34.30%	30.79%
Solution presented, C^4 only	32.74%	**24.53%**	19.66%	25.64%
Solution presented	**26.28%**	25.94%	**18.54%**	**23.59%**

Table 5.1: False Alarm Rate of the solution developed, compared to state of the art algorithms

Sample detections from camera 5 are presented in figure 5.5. In them, the effectiveness of the image processing methods applied can be perceived, as the final solution contains detections that are not present in either C^4 or HOG when they work on their own. On figure 5.6, the final solution discards all false positive detections that HOG would make. On a visual inspection, the bounding boxes created are also more accurate than on both algorithms.

In order to evaluate the impact of using two detection algorithms, the final solution was run with only one of the algorithms functioning at a time, with all the other processing methods in use. It was found that the combination of the HOG and C^4 results provided an average improvement of the final results. The detection rate (table 5.2) raised from 50% and 57%, respectively, to 65% when using both detection algorithms.

As for the processing speed, the solution developed added a very little amount of overhead. The speed at which the sequences were processed can be seen on table 5.4. As sequences from cameras 2 and 7 have 500 frames each, while camera 5 has 768 frames, the results are presented in frames per second, for an equal comparison.

The solution presented performed better than HOG, despite including it in its detections. This is due to the background removal process, which made it faster for the HOG algorithm to process the image with most of its areas masked in black. This type of result also occurred when the solution was run with only the C^4 algorithm, but this cannot be seen on the final program, as the processing time will always skew to the slowest of the algorithms being used.

When using the foreground segments individually, as shown in figure 4.8c, the results were slightly worse, but the improvements in processing speed may reveal useful for some applications. The improvements in speed come from the use of smaller image segments, instead of a full resolution image with portions masked. However, as that constrains the movement of the detection window, a lower number of detections is made.

If the target is to achieve better speeds at the cost of some detections, this alternative method can be used. The improvements are specially noticeable when there are no foreground elements in the scene. In these situations, if the whole image is processed, the detection algorithms will still evaluate a full resolution image, despite being masked out. But if the isolated segments method

Detection Rate				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	51.52%	64.11%	52.90%	56.18%
C^4	27.03%	44.31%	45.17%	38.84%
Solution presented, HOG only	43.56%	58.31%	47.88%	49.92%
Solution presented, C^4 only	44.51%	**69.03**%	57.95%	57.16%
Solution presented	**59.34**%	66.04%	**68.24**%	**64.54**%

Table 5.2: Detection Rate of the solution developed, compared to state of the art algorithms

is used, no detection will even be attempted, as there are no foreground elements to process. This situation can be seen in the speed results from camera 2 (table 5.6), which contains some amount of time with no movement in the scene.

The detection rate and speed results from this usage method can be seen in tables 5.5 and 5.6.

Accuracy				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	56.54%	74.86%	57.43%	62.94%
C^4	27.36%	46.68%	47.10%	40.38%
Solution presented, HOG only	48.33%	65.90%	51.13%	55.12%
Solution presented, C^4 only	45.59%	70.23%	59.70%	58.51%
Solution presented	**65.65**%	71.39%	**73.05**%	**70.03**%

Table 5.3: Accuracy results of the solution developed, compared to state of the art algorithms

Processing speed				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	1.97	1.96	1.98	1.97
C^4	4.14	4.98	4.25	4.46
Solution presented	2.14	2.07	2.14	2.12

Table 5.4: Processing speed of the solution developed, compared to state of the art algorithms (frames per second)

Detection Rate with segments				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	51.52%	64.11%	52.90%	56.18%
C^4	27.03%	44.31%	45.17%	38.84%
Solution presented	46.04%	**65.20**%	**56.55**%	55.93%

Table 5.5: Detection Rate of the solution developed using isolated segments, compared to state of the art algorithms

Processing speed with segments				
Sequence / Detector	Camera 2	Camera 5	Camera 7	Average
HOG	1.97	1.96	1.98	1.97
C^4	4.14	4.98	4.25	4.46
Solution presented	**4.75**	3.31	3.90	3.99

Table 5.6: Processing speed of the solution developed using isolated segments, compared to state of the art algorithms (frames per second)

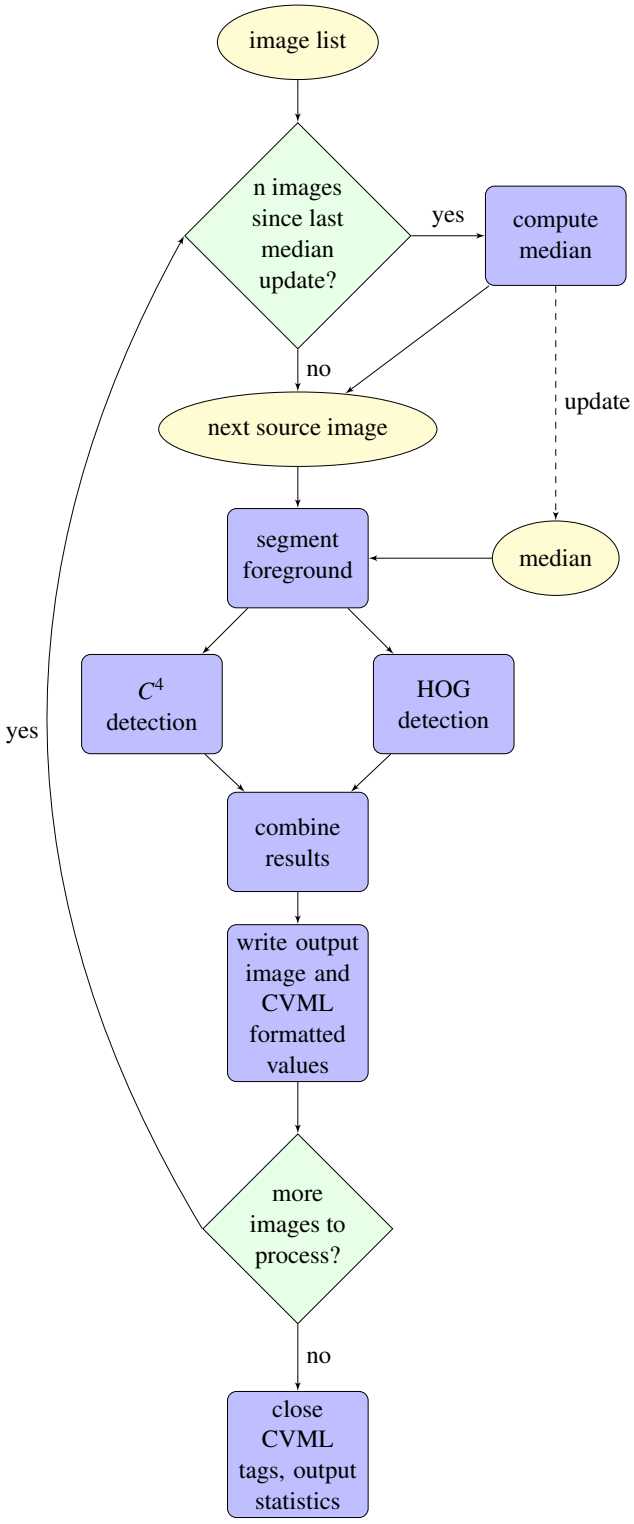


Figure 5.3: Workflow of the developed solution

(a) C^4

(b) HOG

(c) Developed

Figure 5.4: Sample detections from camera 2

(a) C^4

(b) HOG

(c) Developed

Figure 5.5: Sample detections from camera 5

(a) C^4

(b) HOG

(c) Developed

Figure 5.6: Sample detections from camera 7

Chapter 6

Conclusion

6.1 Conclusion

Effective and reliable human detection on still images or video is presently one of the most challenging aspects in the field of computer vision. As such, improvements in this field come in small amounts, as each detection environment presents its own challenges, and a complete, all around perfect solution is still many years away. Yet, all the improvements achieved are still a step towards that goal.

This dissertation presented the interesting challenge of aspiring to improve the results from current solutions in a type of sequences that is rarely addressed in papers: low quality, badly compressed video with a low frame rate. This particular nature of the video brought the focus on its quality, trying to make its processing by the detection algorithms more effective, improving their results. As the results showed, improvements on the detection results were achieved on all metrics tested, and were just as noticeable on the video sequences, by visual inspection.

Various methods of image processing and enhancement were tested, but most were deemed ineffective in this particular case, also revealing the need for better processing tools, as their development also failed to take into account low quality video sources. Even the method that was implemented in the final program had to be written specifically for this application, so the sequences in use could be targeted appropriately. Approaches such as super-resolution and optical flow can provide good results when applied on higher quality video, but they could not be implemented in the recording environments targeted by this dissertation.

A new dataset containing video sequences captured in a retail environment was also created. Those sequences were annotated and will be able to be used as a ground truth reference for new developments in this area. The adapted filming script can also be used as a guide for sequence capturing in other retail stores, in order to emulate the behavior from real retail costumers.

In the end, the solution developed improved on what state of the art implementations could achieve, while being flexible to be configured to handle other sequences and integrated with other algorithms. To maximize results with video recordings that present other characteristics, some configuration parameters can be changed and adapted to handle different video frame rates and

environment variations. It can also be integrated with more detection algorithms, as long as their source code is available, and this combination of more and different algorithms can improve the results even further, making this a good starting point in the development of other human detection pipelines for low quality images. As the results show, the combination of two different algorithms has improved the detection, as by working in a different way, HOG and C^4 end up complimenting each other and making different detections.

6.2 Future work

The solution presented in this dissertation can be enhanced in many ways, by taking advantage of the knowledge of the processing methods that would not work and testing new approaches. Excluding changes to the source sequences (by using a better recording equipment, for example), the most relevant paths of improvement are described here.

The development of a new human detection algorithm targeting the kind of video sequences that were used in this dissertation would probably have the biggest impact of all the possible developments. As the background is eliminated, there is a very high chance that the remaining areas contain at least one person. As such, the focus could be placed on the detection of contours and relevant features in the areas belonging to the foreground, by modifying or creating a feature extraction method that accounted for the low quality of the images.

Since the encoding process made some of the pixels in static areas of the video sequences appear to change from one frame to the next, a method that could prevent the value of those pixels to fluctuate could improve the background removal process, making the foreground extraction much more accurate. This process would need to take into account the foreground elements, as they could overlap those pixels at any time. As such, the effects of the smoothing process could be applied only if the value changes within a very strict range.

The background extraction process could also be enhanced and complemented, either by applying the subpixel smoothing described earlier, or by taking advantage of the multi-camera information. Since the scene presents an overlap between the areas filmed by each camera, the foreground detection masks could be combined between different cameras angles. By using both subpixel smoothing and multi-camera information, the background extraction could make the foreground masks match the moving elements presented in the scene with much more accuracy, lessening the guess work needed in the creation of the bounding boxes around foreground elements.

The current solution requires some level of configuration to provide the best results in each recording environment. Parameters such as the median update rate and number of samples on each update depend mostly on the video frame rate, so an improvement could be done to use the frame rate as an input, and adjust those values accordingly.

Lastly, due to the time it takes to generate the ground truth annotations, other video sequences were not properly tested. It could be interesting to assess the impact of the solution developed on sequences that differ from typical retail environments.

Appendix A

Sequences script

The script presented here is an adaptation of a longer script that takes into account the way costumers behave while shopping in a retail store [33] [34]. It contains 10 scenes in which the actors follow certain actions, which encompass a wide range of consumer behaviors and characteristics, like walking into certain directions and areas of the store, touching products, gestures and conversation. This version was adapted in a way that made feasible its execution, regarding both the time frame available for capturing the sequences, the number of participants, and the store architecture, while keeping the same behavior patterns as the original script.

Legend:
Si - Scene i
Mi – Male actor i
Wi – Female actor i
im - Estimated duration, i minutes

Summary of each scene:

S1
W1 W2
10m
W2 enters store, calmly looking at every product and picking up some of them. Moves to women section, speeds up and only looks at the most interesting things. W1 offers help, which is refused. W2 picks a product and sees how it looks in the mirror. Returns the product to the shelf and leaves, still looking around.

S2
M3 W1
5m
M3 goes directly to preferred zone (around the middle of the store). Doesn't find his size, asks

for help. W1 checks in storage room and tells M3 there isn't stock. Both go to the register, W1 verifies on PC, calls another store and orders product. W1 explains situation to M3, he thanks her and leaves.

S3
M1 M2 W3
10m
M2 goes directly to preferred zone (around the middle of the store) and looks for a product in his size. W3 (wife) enters, looks around and goes to M2. M2 finds product, shows W3 and goes to dressing room. M2 exits dressing room, shows W3 how product fits, they approve and M2 enters dressing room again. W3 searches for something for herself. M2 exits dressing room and helps W3 find a product. After a while, they give up, go to register and buy the product. While M2 pays M1, W3 leaves and then M2 leaves.

S4
W2 W3
5m
W2 and W3 (friends) enter shop in a happy mood, looking around and picking up what looks interesting. They stay together and ask for opinion most times they try something. They enjoy trying the products. After a while they leave without buying anything.

S5
M2 M3 W1 W3
8 10m
W3, M2 (husband), M3 (son) enter store. W3 knows what she wants and looks for it, following store cues. M2 and M3 follow her, side by side, talking and uninterested in the store. W3 finds product, they all stop, she likes it and they approve. W3 keeps the product and asks M3 to help her find more things. M2 goes alone to the men section and picks some products. After a while, M2 goes to W3 and M3. They are accompanied by W1, as W3 asked for help. W3 stays with W1 and gives products to M2. M2 and M3 go to queue to pay. After a while, W3 goes to queue too. They wait a little, W3 pays and they all leave.

S6
M2 M3 M4 M5 W1
5 7m
All men enter slowly, in pairs, talking face to face. M5 and M2 split slowly, end conversation and start looking for products individually. M3 and M4 stop near the register to finish the conversation. M4 starts looking at the store while talking. After a while, the conversation stops and M4 starts to explore the shop. M3 stays in place. M4 joins M5 and talks about products seen. M2 joins M4 and M5 and asks for opinion on a product he brings. They like it, M2 goes to empty register, pays

W1 and stays and talks with M3. M4 and M5 join them and they all leave.

S7
M4 W1
5 7m
M4 enters shop, looking for something but disoriented. W1 realizes M4 needs help and offers it. M4 accepts and W1 starts showing products that might interest him. M4 goes to the dressing room twice and after leaving delivers the product to W1, who stores it in the register. At the end, they go to the register, M4 pays and leaves with the two products.

S8
M2 W3
5m
M2 and W3 (couple) enter store. They look to only be comparing prices. They stay together through both male and female shopping zones. After a while they leave.

S9
M3 W1
5m
M3 enters shop. Looks around and picks some products quickly. W1 approaches and tries to attract him to their promotions. "It depends on the persuasive characteristics of the shop-assistant, and on the promotions, if the man buy something or not. The man normally is open to promotions."

S10
M1 M4 W2
5m
W2 enters shop in a hurry. She asks M1 for assistance and explains what product she wants (type, destination, price). M1 and W2 visit 3 zones. W2 brings 2 products and goes to the queue, where M4 is. W2 asks M4 if she could get in front, he allows it. W2 pays and leaves.

S11
M2 M4 W1 W2 + extras for queue
10m
M2 and W2 (couple) enter store with M4 (friend). W2 leads, and goes to female zone. M2 and M4 go to the men section, look around individually but stay close. W2 calls and M2 goes to her. W2 shows a product to M2, he doesn't like it and helps her find another. After several tries, they agree on a product. W2 goes to the dressing room, while M2 returns to M4, in the men section. M2 and M4 talk about products M4 picked. W2 arrives very happy and goes to the queue with M2. M4 tries a last product, likes it and goes directly to where M2 and W2 are in the queue. The other people are angry at M4. M2 and W2 try to help, until W1 (shop employee) makes M4 go

to the end of the queue. M2 and W2 pay and wait for M4. They all leave, with M4 a little annoyed.

Time (m)	0	5	10	15	20	25	30	35	40
Scenes	S1: W1 W2		S5: M2 M3 W1 W3+queue		S2: M3 W1	S6: M2 M3 M4 M5 W1		S9: M3 W1	
	S3: M1 M2 W3		S7: M4 W1	S10: M1 M4 W2	S8: M2 W3	S4: W2 W3		S11: M2 M4 W1 W2+queue	
Total	2M 3W	2M 3W	3M 3W	4M 3W	2M 2W	4M 3W	4M 3W	3M 3W	2M 2W

Figure A.1: Timetable. Est. 45m. Total people: 4 Men 3 Women

References

[1] OpenCV. How to use background subtraction methods, 1 2014. URL: http://docs.opencv.org/trunk/doc/tutorials/video/background_subtraction/background_subtraction.html.

[2] Jianxin Wu and J.M. Rehg. Centrist: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2009. doi:10.1109/TPAMI.2010.224.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, 2005.

[4] Jianhao Ding, Yigang Wang, and Weidong Geng. An hog-ct human detector with histogram-based search. *Multimedia Tools Appl.*, 63(3):791–807, April 2011. URL: http://dx.doi.org/10.1007/s11042-011-0896-9, doi:10.1007/s11042-011-0896-9.

[5] Duc Thanh Nguyen, Philip Ogunbona, and Wanqing Li. Human detection with contour-based local motion binary patterns. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3609–3612, 2011.

[6] Jianxin Wu, C. Geyer, and J.M. Rehg. Real-time human detection using contour cues. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 860–867, 2011. URL: https://sites.google.com/site/wujx2001/home/c4, doi:10.1109/ICRA.2011.5980437.

[7] TRECVID. Trec video retrieval evaluation: Trecvid, 01 2014. URL: http://trecvid.nist.gov/.

[8] The Guardian. Tesco face detection sparks needless surveillance panic, facebook fails with teens, doubts over google+, 11 2013. URL: http://gu.com/p/3k8ye.

[9] Douglas McCormick. Face recognition failed to find boston bombers, April 2013. URL: http://spectrum.ieee.org/riskfactor/computing/networks/face-recognition-failed-to-find-boston-bombers.

[10] CCTV Camera Pros. Hd-d20 hd security camera, dome hd-sdi cctv surveillance camera. URL: http://www.cctvcamerapros.com/HD-Security-Camera-p/hd-d20.htm.

[11] BBC. Agnes sina-inakoju murder: Cctv of takeaway killer. URL: http://www.bbc.co.uk/news/uk-england-london-13051697.

[12] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

[13] Mao-Hsiung Hung, Jeng-Shyang Pan, and Chaur-Heh Hsieh. A fast algorithm of temporal median filter for background subtraction. 2014.

[14] Tang Shaopeng and S. Goto. Histogram of template for human detection. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2186–2189, 2010.

[15] Patrick Ott and Mark Everingham. Implicit color segmentation features for pedestrian and object detection. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 723–730. IEEE, 2009.

[16] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

[17] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer Vision-ECCV 2004*, pages 469–481. Springer, 2004.

[18] Timo Ojala, Matti Pietikainen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[19] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[20] Gustaf Kylberg and Ida-Maria Sintorn. Evaluation of noise robustness for local binary pattern descriptors in texture classification. *EURASIP Journal on Image and Video Processing*, 2013(1):1–20, 2012. URL: http://dx.doi.org/10.1186/1687-5281-2013-17, doi:10.1186/1687-5281-2013-17.

[21] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE, 2009.

[22] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision—ECCV'94*, pages 151–158. Springer, 1994.

[23] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[24] B. Sirmacek and C. Unsalan. Urban-area and building detection using sift keypoints and graph theory. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(4):1156–1167, April 2009. doi:10.1109/TGRS.2008.2008440.

[25] Stephen Se, David Lowe, and Jim Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 2051–2058. IEEE, 2001.

[26] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*, pages 589–600. Springer, 2006.

[27] Fatih Porikli and Tekin Kocak. Robust license plate detection using covariance descriptor in a neural network framework. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 107–107. IEEE, 2006.

[28] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[29] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[30] T Huang, G Yang, and G Tang. A fast two-dimensional median filtering algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(1):13–18, 1979.

[31] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[32] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734–741 vol.2, 2003.

[33] BJ Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 40. ACM, 2009.

[34] Na Li and Ping Zhang. Consumer online shopping attitudes and behavior: An assessment of research. In *Eighth Americas Conference on Information Systems*, pages 508–517, 2002.

[35] Thor List and Robert B. Fisher. Cvml - an xml-based computer vision markup language. In *ICPR (1)*, pages 789–792, 2004. URL: http://dblp.uni-trier.de/db/conf/icpr/icpr2004-1.html#ListF04.

[36] Tim Ellis. Performance metrics and methods for tracking in surveillance. In *Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'02)*, pages 26–31. Citeseer, 2002.

[37] James Black, Tim Ellis, and Paul Rosin. A novel method for video tracking performance evaluation. In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 125–132, 2003.

[38] Faisal Bashir and Fatih Porikli. Performance evaluation of object detection and tracking systems. *In PETS*, 6, 2006.

[39] Intel. Intel® core™ i7-4700mq processor. URL: http://ark.intel.com/products/75117.

[40] Michael E Tipping and Christopher M Bishop. Bayesian image super-resolution. In *Advances in neural information processing systems*, pages 1279–1286, 2002. URL: research.microsoft.com/pubs/67152/bishop-nips02-superres.pdf.

[41] Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *Image processing, IEEE Transactions on*, 13(10):1327–1344, 2004.

[42] OpenCV. Eroding and dilating, 12 2013. URL: http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html.

[43] Tang Shaopeng and S. Goto. Human detection using motion and appearance based feature. In *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, pages 1–4, 2009.

[44] ffdshow. Directshow and vfw codec for decoding/encoding using libavcodec, xvid and other opensourced libraries with a rich set of postprocessing filters. URL: http://sourceforge.net/projects/ffdshow/.

[45] Murtaza Taj and Andrea Cavallaro. Multi-camera track-before-detect. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pages 1–6. IEEE, 2009.

[46] OpenCV. Image moments, 12 2013. URL: http://docs.opencv.org/doc/tutorials/imgproc/shapedescriptors/moments/moments.html.

[47] OpenCV. What is the carttopolar angle direction in opencv?, 4 2013. URL: http://answers.opencv.org/question/11006/what-is-the-carttopolar-angle-direction-in-opencv/.