

**Faculdade de Engenharia da Universidade do Porto**



**Image Descriptors for Counting People with  
Uncalibrated Cameras**

João Vasco Dantas dos Reis

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Ph.D. Daniel Moura  
Co-supervisor: Eng. Eduardo Marques

July 30, 2014



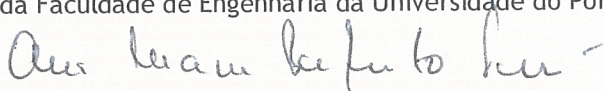
A Dissertação intitulada


“Image Descriptors for People Counting with Uncalibrated Cameras”

foi aprovada em provas realizadas em 18-07-2014

o júri

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

  
Professora Doutora Ana Maria Perfeito Tomé  
Professora Associada do Departamento de Eletrónica, Telecomunicações e  
Informática da Universidade de Aveiro

  
Doutor Daniel Cardoso de Moura  
Investigador Auxiliar do Departamento de Engenharia Eletrotécnica e de  
Computadores da Faculdade de Engenharia da Universidade do Porto

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 - João Vasco Dantas dos Reis

Faculdade de Engenharia da Universidade do Porto



# Resumo

Nesta dissertação, é feito um estudo acerca de descritores de imagem para contagem de pessoas em ambientes urbanos. O algoritmo de contagem desenvolvido não requer calibração da camera de vídeo, pois usa um mapa com informação de escala para pesar os pixels da imagem, de modo a tornar o método invariante à perspectiva. Este mapa foi também utilizado para modificar um descritor notável na área de detecção humana, da autoria de N.Dalal e B.Triggs, de modo a torná-lo mais robusto para contagem de pessoas. Assim, é proposto um novo descritor denominado *Perspective invariant Histograms of Oriented Gradients (HOGp)*. A relação entre as características dos histogramas e o número de pessoas presentes na imagem permite inferir contagens para novas imagens, através de modelos de regressão. Os resultados experimentais com os datasets UCSD e FC demonstram o potencial do método seguido bem como o valor do descritor proposto.



# Abstract

In this thesis is made a study on image descriptors for people counting in urban environments. The proposed counting algorithm does not require camera calibration. Instead it uses a map with scale information to weight image pixels, in order to make the algorithm perspective invariant. This map was also used to extend a remarkable descriptor in human detection, proposed by N.Dalal and B.Triggs, making it more robust for people counting purposes. Therefore, in this thesis is proposed a new image descriptor that was called *Perspective invariant Histograms of Oriented Gradients (HOGp)*. The relation between histograms' features and the number of people in the image allows the counting inference for new images, by using regression based models. Experimental results with UCSD and FC datasets demonstrate the potential of the followed method as well as the value of the proposed descriptor.





# Agradecimentos

Aos orientadores, Professor Daniel Moura e Eng. Eduardo Marques, pela motivação, dedicação e partilha de conhecimento que me deram desde o início até ao fim deste percurso, sem as quais acredito que não teria sido possível concluir esta tarefa da mesma forma. Ao Eng. Pedro Cunha, pela disponibilidade, já que, não sendo um orientador *per se*, foi um fonte de ajuda inesgotável.

Aos meus amigos. Aos meus irmãos. À minha família, a quem devo grande parte do que sou.

Aos meus pais.

João Vasco Reis



*“If I have seen further than others, it is by standing upon the shoulders of giants.”*

Isaac Newton



# Contents

<b>Introduction.....</b>	<b>1</b>
1.1 Motivation.....	2
1.2 Context .....	2
1.3 Objectives.....	3
1.4 Document outline.....	3
<b>State of the Art .....</b>	<b>5</b>
2.1 Trajectory clustering.....	6
2.2 Feature-based regression.....	7
2.3 Individual pedestrian detection .....	10
2.4 Other approaches .....	12
2.5 Summary.....	13
<b>Image descriptors for people counting.....</b>	<b>15</b>
3.1 Dataset selection .....	16
3.2 Background Subtraction .....	16
3.3 Perspective normalization .....	18
3.4 Ground truth annotation .....	20
3.5 Feature extraction.....	21
3.6 Perspective invariant histograms of oriented gradient .....	24
3.7 Regression Models .....	25
3.8 Evaluation metrics.....	27
3.9 Summary.....	27
<b>Results .....</b>	<b>29</b>
4.1 Ranking of image descriptors.....	29
4.2 Sensitivity analysis of the descriptors' parameters.....	30
4.3 Evaluation of different combinations of descriptors .....	34
4.4 Impact of training set size .....	36
4.5 Evaluation on the Future Cities dataset .....	37
4.6 Discussion .....	39
<b>Conclusions and Future work.....</b>	<b>41</b>
<b>Appendix A – Error tables .....</b>	<b>43</b>
A1 MAE for UCSD (631:1:1400 as training set) .....	43
A2 MAE for UCSD (635:5:1400 as training set) .....	44
A3 MAE for UCSD (640:80:1360 as training set).....	45
A4 MAE values for FC (first 25 frames as training set) .....	46
A5 MAE values for FC (first 50 frames as training set) .....	47
A6 MAE values for FC (first 100 frames as train) .....	48
<b>References .....</b>	<b>49</b>



# List of Figures

<b>Figure 2.1</b> – Example of stacks used on [4]. Horizontal axis corresponds to horizontal dimension of the original images. Vertical axis corresponds to time in the original sequence. Source: extracted from [4].	7
<b>Figure 2.2</b> – Features used in [10]: (a) original image, (b) foreground mask image, (c) edge detection map, (d) the edge map after de "AND" operation between (b) and (c). Source: extracted from [10].	8
<b>Figure 2.3</b> – Crowd counting results for site A with 30° camera angle, from [10]. Above – fitting with histograms. Below – fitting without histograms. Source: extracted from [10].	8
<b>Figure 2.4</b> – Flowchart of the whole system from [9]. Source: extracted from [9].	9
<b>Figure 2.5</b> – Crowd counting system: the scene is segmented into crowds with different motions. Normalized features that account for perspective are extracted for each segment and the crowd count for each segment is estimated with a Gaussian process. Source: extracted from [18].	10
<b>Figure 2.6</b> – A sequence of images showing critical cases of blob splitting, merging and displacement. Source: extracted from [23].	11
<b>Figure 2.7</b> – The counting people system from [11]. (a) Input image, (b) Background subtraction, (c) Skeleton graph, (d) Head detection and pose estimation, (e) Head tracking. Source: extracted from [11].	12
<b>Figure 3.1</b> – Background Subtraction algorithm. (a) Example frame of the UCSD dataset. The ROI for this scene is marked by the green line. Yellow box marks the considered blob for (c) and (d). (b) Corresponding foreground mask with ROI obtained by the BS algorithm. (c) Foreground submask for one segmented blob. (d) The same blob after dilate morphological operation.	18
<b>Figure 3.2</b> – Example of images for feature extraction. (a) Foreground binary mask. Yellow line marks blob Bounding Box. (b) Grayscale image for the corresponding blob of (a).	18
<b>Figure 3.3</b> – Perspective map for the UCSD dataset. a) Reference person at the closer extreme of the scene, and (b) at the distant extreme. b) The perspective map which scales pixels by their relative size in the true 3D scene. Source: extracted from [14].	19
<b>Figure 3.4</b> – Ground truth annotation process. Manual annotations (left) are overlayed on the foreground segmented objects (centre), and the region overlaps are used to automatically determine ground truth counts for each blob (right). Source: adapted from [45].	20
<b>Figure 3.5</b> – HOGp simulation. Bottom image has half the blob area of top image. It is observable that, for a perfectly resized image, the flat histogram bins from HOGp match the values for the original image, while HOG does not.	25

<b>Figure 4.1</b> – Experimental results of window size influence on HOG and HOGp descriptors with fixed 8 bins. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing. ....	31
<b>Figure 4.2</b> – Experimental results of window size influence on HOG and HOGp descriptors with fixed 16 bins. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing. ....	32
<b>Figure 4.3</b> – Experimental results of number of bins influence on HOG and HOGp descriptors with fixed 1x2 window size. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing. ....	33
<b>Figure 4.4</b> – Experimental results of number of bins influence on HOG and HOGp descriptors with fixed 2x1 window size. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing. ....	33
<b>Figure 4.5</b> – Experimental results of Polynomial order variation on Zernike moments descriptor reported by Linear Regression model. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing. ....	34
<b>Figure 4.6</b> – People counting results on UCSD dataset using HOGp and Zernike as image descriptor and Linear Regression. Frames 631-1400 were set aside for training, and frames 31-600 and 1401-2000 were used for testing. ....	36
<b>Figure 4.7</b> – Experimental results on UCSD dataset with 3 different training subsets: Full, 635:5:1400 and 640:80:1360. Frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. ....	37
<b>Figure 4.8</b> – Experimental results on FC dataset with 3 different training subsets: first 25 frames, 50 frames and the first 100 frames. The remaining dataset frames were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. ....	38
<b>Figure 4.9</b> – People counting results on FC Dataset using HOGp, Area and Perimeter as image descriptors and Linear Regression with Bagging. The first 100 frames were set aside for training and the remaining were used for testing. ....	38



# List of tables

<b>Table 3.1</b> – Nine dataset were considered to develop the proposed crowd counting algorithm. A subset of the total number frames was annotated at constant frame spacing indicated by the column interval. FC dataset was constructed on a later development stage of the thesis. ....	16
<b>Table 3.2</b> – Parameter values used in MOG2 method, for both UCSD and FC dataset.....	17
<b>Table 3.3</b> – Considered regions treated as sets of pixels using set notation.....	21
<b>Table 4.1</b> – Relief-F feature selection algorithm output for the UCSD Crowd Counting Dataset. Ranking results for 15 image descriptors with different number of features, for a total of 458 features. Feature attributes are grouped by their respective image descriptor and ordered by best to worst score. ....	30
<b>Table 4.2</b> – Experimental results on UCSD dataset. Frames 631-1400 were set aside for training, and frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by 9 Regression Models. The lowest MAE value for each Regression Model is underlined and the lowest value of all is bolded.....	35
<b>Table 4.3</b> – Experimental results on UCSD dataset with 3 different training subsets: Full, 635:5:1400 and 640:80:1360. Frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. The lowest MAE value for each Regression Model is underlined and the lowest value of all is bolded. ....	36
<b>Table 4.4</b> – Experimental results on FC dataset with 3 different training subsets: first 25 frames, 50 frames and the first 100 frames. The remaining dataset frames were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. The lowest MAE value for each Regression Model is underlined and the lowest value of all is bolded.....	37
<b>Table 4.5</b> – Testing results on the UCSD dataset. Frames 1-600 and 1401-200 were used for testing. Results underlined correspond to the proposed approach. ....	40



# Abbreviations

BS	Background Subtraction
EKF	Extended Kalman Filter
FC	Future Cities
FCUP	Faculdade de Ciências da Universidade do Porto
FEUP	Faculdade de Engenharia da Universidade do Porto
FPCEUP	Faculdade de Psicologia e de Ciências da Educação da Universidade do Porto
GLCM	Gray Level Co-occurrence Matrix
GLDM	Gray Level Difference Matrix
GLRL	Gray Level Run Length
GMM	Gaussian Mixture Model
HOG	Histogram of Oriented Gradients
HOGp	Perspective invariant Histograms of Oriented Gradients
HSV	Hue Saturation Value
MAE	Mean Absolute Error
MRF	Markov Random Field
OS	Operative System
PETS	Performance Evaluation of Tracking and Surveillance
QUT	Queensland University of Technology
RGB	Red Green Blue
ROI	Region Of Interest
UCSD	University of California, San Diego
UP	Universidade do Porto
WIP	Work In Progress



# Chapter 1

## Introduction

The ability to know the number of people crossing a determined space has always emerged interest in several application areas such as transportation, commerce, education, robotics, etc. Creating resources that allow this type of data acquisition is a current engineering challenge, as it can be made using different sensor systems in order to achieve almost the same results.

In recent times, the usage of vision based systems has gained importance and visibility across many applications, and R&D groups are increasing their resources on developing technology based on this type of sensors. This is mainly due to its vast applicability and flexibility allied with a relative low-cost, when compared to other sensor solutions.

Using image processing techniques, the input data acquired from a video camera can be processed according to users' interest, producing output results in the desired format. This processing chain brings up many challenges for the developer. For instance, the input data is a sequence of images with high noise values and irrelevant data elements. Also, the image processing algorithms may have to adjust to different environments and, in some cases, learn and adapt.

Automated people counting has become an active field of computer vision research in recent years (e.g. [1]–[3]). From a technological standpoint, computer vision solutions typically focus on detecting, tracking, and analyzing individuals. These approaches are not scalable for people counting on urban outdoor scenarios. With large and dense crowd sizes, where occlusion happens frequently and each pedestrian is depicted by a few image pixels, both individual detection and tracking becomes a nearly impossible task. Furthermore, if the solution is based on individual detection and tracking, the computational requirements and system's cost are elevated.

Other approaches focus on analyzing images as whole, holistically or globally. These methods rely on crowd properties or individual's deviation from the crowd, to estimate the number of people that constitute said crowd. Although a number of holistic approaches have been proposed, its viability for outdoor environments has not been fully established, due to the inability to control scene illumination, crowd dynamics and crowd density.

The method that is followed in this thesis is different from both the “individual-centric” and “crowd-centric” approaches. Crowd is segmented into individuals or groups of people, and each segment is analyzed separately, in order to infer the number of people that constitutes each segment. These local estimations are summed to retrieve the total crowd size of a given image.

Segment analysis is done using image descriptors, which can be described as methods to extract characteristics, statistics and features of an input image. In this work, several low-level image descriptors are

used, in order to determine which ones are best suited to describe image segments for people counting retrieval. Furthermore, the cameras that are used in this work are not calibrated, so it is proposed an extension to an image descriptor that has remarkable value in human detection, with the objective to account for perspective effects in its calculation.

## 1.1 Motivation

Crowd counting and density estimation contributes to crowd management for safety and surveillance such as deployment of law enforcement activity and unusual behavior detection. It is also helpful in finding the number of pedestrians or commuters, which can be important for planning and developing public infrastructures. Furthermore, it can be used to gauge political significance of rallies and social impact of cultural events.

In order to calibrate a video-camera the user must have access to camera specifications. This can be a problem if the video is, for instance, historical, or simply because there is no way to know camera model. In this thesis this problem is addressed by focusing on an algorithm that doesn't rely on video camera calibration.

Privacy is a major theme in this dissertation. When the technology involves video recording or streaming of human individuals it usually brings up a lot of skepticism among persons. Also, video-streaming requires a significant bandwidth or high cost infra-structures. If the developed work manages to perform well with low quality videos, where individuals cannot be recognized, some of these inherited privacy concerns would be surpassed.

Furthermore, using a single camera per scene and focusing on algorithms that don't require high processing power or memory, this work could open the possibility to be implemented as an embedded solution, which could count people on-site and, for instance, send exclusively statistical data on crowd density to a network.

This dissertation is driven by the chance to overcome some of the inherited difficulties in counting people with computer vision, creating a low cost, low resource, privacy preserving system that can be used in different urban environments.

## 1.2 Context

The proposed dissertation is being developed under the Future Cities Project [4], a Seventh Framework Programme (FP7) [5] funded project. This is a WIP project by Universidade do Porto (UP) that involves research groups from several faculties such as Engineering (FEUP), Psychology (FPCEUP) and Sciences (FCUP). The main goal is to turn the city of Porto into an urban-scale living lab, creating resources that can be used by companies, start-ups and researchers, to develop and test their technologies, products and services.

In order to achieve the desired goal, several data acquisition systems will be installed in strategic areas of the city as well as mobile spots such as STCP buses. These sensor based systems form a network that can upload data into a cloud service or other database formats. Using the data provided by the network, future users will be able to develop technology based on urban information and statistics.

The Future Cities Project aims to reverse the deterioration of urban cities' facilities and to improve citizens' quality of life. By working with teams from different scientific areas, all focused on the same goals, this project strengthens the overall interdisciplinary research in Portugal.

## 1.3 Objectives

The main goal of the dissertation is to design, implement and test a system to detect and count the number of people on outdoor urban environments. The following objectives are expected to be accomplished by the end this dissertation:

- O1.** To design a vision-based algorithm for counting people.
- O2.** To design a method to adapt the algorithm for uncalibrated cameras.
- O3.** To evaluate different image descriptors for people counting purposes.
- O4.** To propose an image descriptor focused on people counting in urban environments.
- O5.** To evaluate different regression based models for people counting systems.

## 1.4 Document outline

This chapter provides a brief background of the Future Cities Project, describes the thesis motivation, main challenges and objectives. Chapter 2 presents the State of the Art on the dissertation theme. Chapter 3 describes the followed methodology for achieving the desired objectives. Chapter 4 presents the obtained results and a final discussion on said results. Finally, Chapter 5 presents the conclusion for the proposed thesis and possible future work extension.





## Chapter 2

# State of the Art

Work on the problem of counting people using computer technology ranges from prototype systems tested only in controlled labs to full functional solutions that are used in real-world environments.

The Spanish Railway Company did a market survey of several methods for counting people as reported by Albiol et al. [6]:

- *Mechanical counters*, such as those used to validate the tickets on a train station, are the most accurate, but they require specific environment customization and they involve a physical obstacle to the person passing through.
- *Light beams* are fairly accurate, as long as only one person blocks the beam at a given time. In other words, this method doesn't do well with crowd counting and occlusion, which are very common in urban everyday life. Also it is not appropriate to determine the direction of passing, which might be of valuable knowledge, for example to know exactly the number of people going in and out of a certain building.
- *Differential weight* system is another counting method that uses load cells in order to evaluate the weight variations at a given space and count the number of people from those variations. This is especially useful for carriage environments, like subway or train, because the cells can be installed at the carriage suspension.
- *Sensitive carpet* is another accurate system for counting people. However, sensorial carpets are prone to wear and they involve severe modifications of the "counting" environment.
- *Computer vision* based systems for counting people offer an alternative to these methods. They provide the necessary accuracy without heavy modification of the environment or impeding traffic. Also, this method generally solves the problem at a much lower cost, when compared to the methods described above.

One of the main issue with computer vision systems is the need to separate objects of interest from the remaining pixels. Several proposed counting systems, like the ones described in [7]–[9], use two cameras to help with this process, a method also known as stereovision. Another common procedure is to use background subtraction, as seen for example in [9]–[12]. Using this method, the acquired images are subtracted to a previously saved background of the same capture scene and the resulting image contains the pixels of newer objects which might be people. In uncontrolled environments, such as outdoor urban areas, the background should be updated in function of natural light variations and other climatic factors. Also, recording successive background images enables the detection of permanent background changes, like an object that has left the environment. Fewer systems use other approaches that don't have the need to separate people from the rest of the image, some of which will be discussed later on this chapter.

Another major problem in people detection and counting is the inherent occlusion that increases as the number of individuals in the scene rises. Occlusion consists on the visual obstruction of objects of interest, from the camera view-point. For example, occlusion happens when a group of people is moving within the scene and the individuals leading the crowd block camera vision from the individuals on the back. One good starting point to handle this problem, is to choose a proper camera position according to the counting environment. However, in most of the cases this is not sufficient, and the algorithms should include occlusion handling.

As stated by many authors such as Aziz et al. [13] or Chan et al. [14], various methods that estimate the number of people in input image sequences have been proposed and they can be divided into three main paradigms:

- Trajectory clustering approach
- Feature-based regression approach
- Individual pedestrian detection

## 2.1 Trajectory clustering

In this approach people are counted by tracking and identifying visual features over time. The resulting feature trajectories exhibit coherent motion over time, which enables the opportunity to cluster them and estimate the number of people based on the number of clusters. The main downside of this approach is that real-time processing is nearly impossible. However, the system could record trajectories in short periods of time, while analyzing the last saved clusters, achieving a minor time offset in comparison with real-time systems.

Using stereo differencing and an overhead camera view, Terada et al. [7] developed a system that can count people and determine the direction of movement as they cross a measurement line. The top-down view avoids the occlusion problem, when groups of people cross the camera's field of view. Their system generates space-time images to help determine directionality. However, they only tested the system in a controlled environment – a lobby of an office building – with a small sample of 43 people and no error data was given. Additionally, occlusion is dealt with by requiring the specific overhead top-down camera angle.

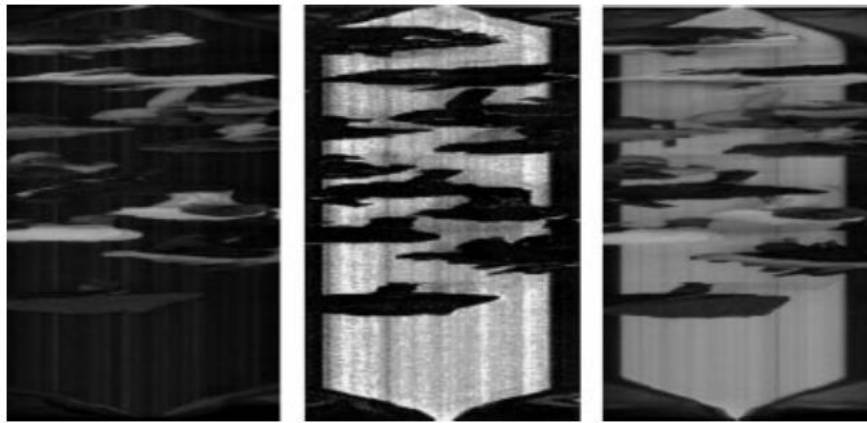
Beymer and Konolige [8] relax the camera position restrictions of Terada et al. [7], while using the same stereo vision approach. To handle occlusion problems, their system uses continuous tracking and detection. This method is able to drop detection of a person whenever he becomes occluded but, if this person returns to the capture field, the system acknowledges this as a new instance, leading to duplicates and counting multiple times the same person. As a matter of fact, the performance of this system drops significantly as the number of people and occlusions increase in a scene. With a small test setup of 5 people and 28 occlusions they achieved a 70% tracking rate.

Sexton et al. [15] use a simplified algorithm for counting people. The usual overhead camera position was used, in order to deal with occlusion. They constantly update the background reference, used for background subtraction and consequent people segmentation and detection. Then, the resulting blobs are tracked frame by frame, simply matching each blob to its closest one, using centroid feature. This system was tested on a Parisian railway station, and they achieved results with counting accuracy ranging from 79% to 99%. Larger crowds cause processing speed to drop, leading to higher error results. Newer, faster processors in today use would probably handle these situations with a much higher performance.

Segen [16] uses a similar approach as Sexton's [15], based on feature extraction and path generation. He uses standard background segmentation techniques to identify areas of interest. Then it extracts features from these blobs and track those features across moving frames. This generates feature paths that are merged into

clusters, representing the movement of a person over time. Those paths can be used to count the number of people that cross a measurement line and their directionality. However, experimentation was lackluster, as the system was only tested to run in real-time with up to 8 people in the scene. Additionally, this system doesn't make any attempt to deal with occlusions.

Albiol et al. [6] developed a system to count people entering and leaving a Spanish public train. The camera was fixed above the door mechanism on the train itself. When the door opens, the system notices a sudden change in light and starts capturing images that are reduced into stacks. These stacks represent space-time images and some examples are illustrated in Figure 2.1. The horizontal axis of the stack has the same horizontal dimension of the original images, while the vertical axis of the stack represents time. To perform segmentation from the background, they use a gradient function instead of background subtraction. Once the door closes, the system starts analyzing the stored stacks as the train moves to the next station. A trained algorithm uses these stacks to determine how many people crossed the threshold of the door. To complete the process, an optical flow algorithm was used to determine direction of passing. With over 149 test stops, this system counted 318 incoming passengers and 379 outgoing passengers when the real was 321 and 385 respectively for an overall accuracy of 98.7%.



**Figure 2.1** – Example of stacks used on [6]. Horizontal axis corresponds to horizontal dimension of the original images. Vertical axis corresponds to time in the original sequence. Source: extracted from [6].

## 2.2 Feature-based regression

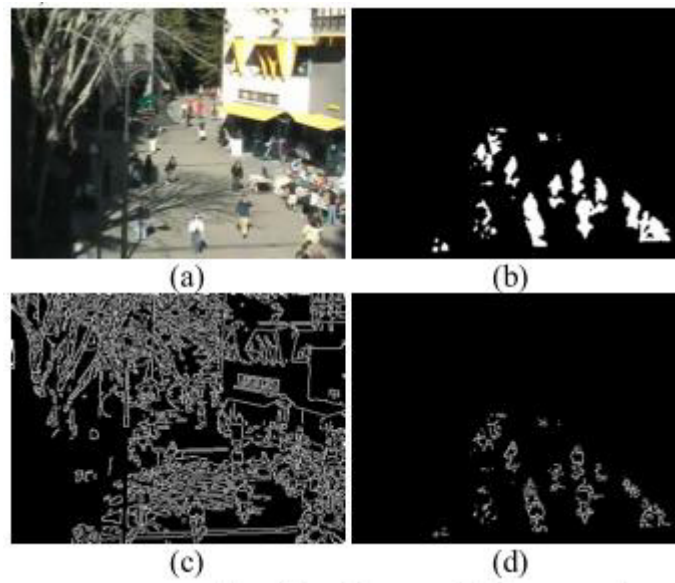
This approach estimates the number of people by a regression of features previously extracted from input images. Some examples of common regression models are neural networks, linear and piece-wise linear. These methods usually are divided into stages, being:

1. Background subtraction
2. Feature extraction
3. Estimating the crowd density or count by a regression function

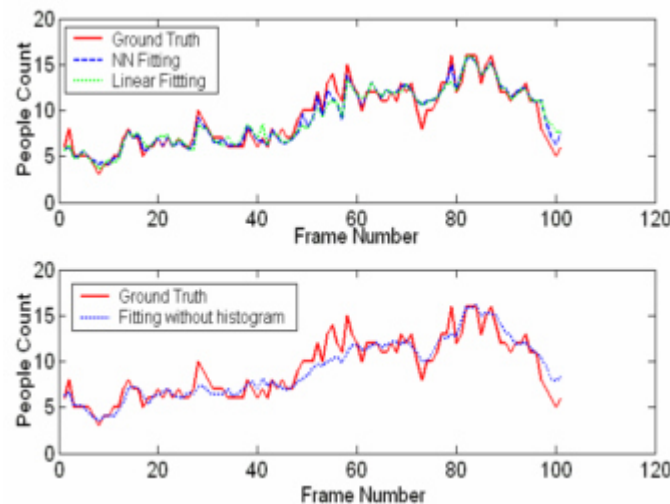
Although feature-based regression first appeared on a subway platform monitoring, in the recent days it has also been applied to outdoor environments.

Kong et al. [12] developed a view-point invariant system to count pedestrians, which can be deployed with minimal setup. The method starts with feature extraction and normalization, which include foreground regions given by a background subtraction algorithm and the edge orientation map generated by an edge detector. They compute a blob size histogram using foreground masks generated for each frame with an adaptive background mixture modeling, developed by Stauffer et al. [17]. Combining this histogram with the edge orientation

histogram they achieve a much “cleaner” image, as illustrated in Figure 2.2. For density estimation, they assume that all the pedestrians in the scene have similar size and that they all lie on a horizontal ground plane. Then, they calculate the ROI and its weights density map, using *homography* allied with relative pixel density calculation. Additionally, the algorithm does feature normalization in order to give a measure of the features that is approximately invariant to the translations of pedestrians on the ground plane and under different camera viewpoints. The system was trained offline to find the relationship between the features and the number of pedestrians in the image, with a method based on neural networks. Experimentation was conducted under two different camera orientations, within two different sites. The experimental results demonstrate the reliability and accuracy of the system, which proves the importance of using feature histograms instead of raw edge and blob features, as illustrated in Figure 2.3.



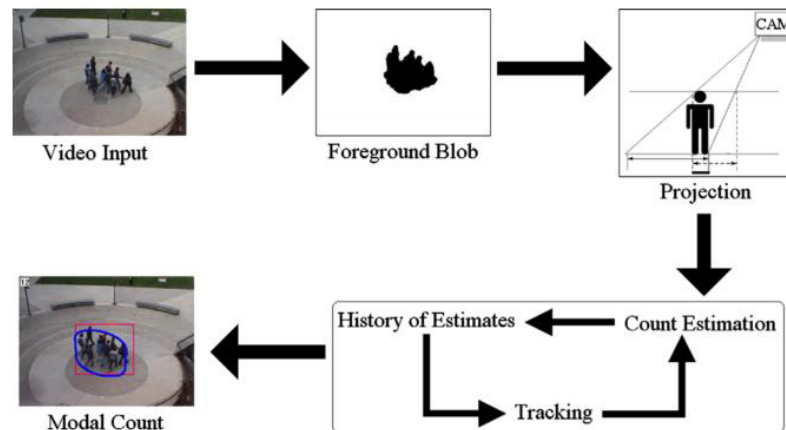
**Figure 2.2** – Features used in [12]: (a) original image, (b) foreground mask image, (c) edge detection map, (d) the edge map after de "AND" operation between (b) and (c). Source: extracted from [12].



**Figure 2.3** – Crowd counting results for site A with 30° camera angle, from [12]. Above – fitting with histograms. Below – fitting without histograms. Source: extracted from [12].

Kilambi et al. [11] propose a method that tackles people counting in a group-based approach, without constraining themselves to detection of individuals. A flowchart of the whole system can be seen in Figure 2.4.

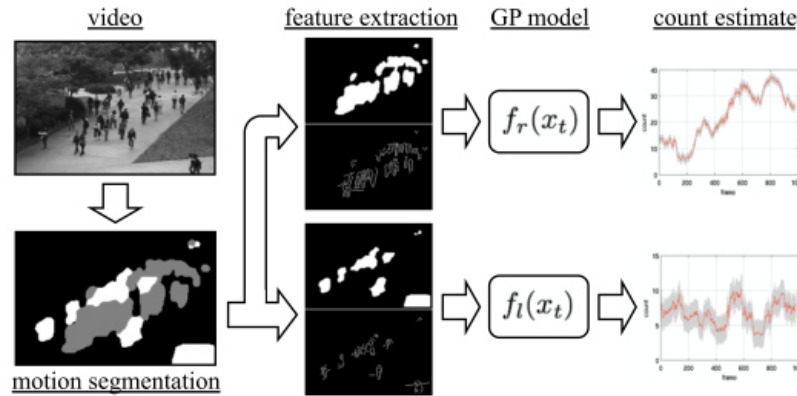
They start by segmenting the foreground regions using the adaptive mixture of Gaussian method proposed by Atev et al. [18]. Objects shorter than a human being are filtered, through projections on the real world ground plane. They track the foreground regions using and EKF pedestrian tracker proposed by Masoud and Papanikolopoulos [19], in order to classify these regions as either individuals, groups, or vehicles. Two methods were used to estimate the number of people in each tracked group. The first consists in a Heuristic-based method, which uses the area occupied by the projections on the ground plane in conjunction with previous algorithm training. This method provides a simple yet efficient solution to group counting. However it performs poorly when the groups are more spread out, when the group configuration or dynamics changes and when the height of a group differs significantly from the fixed head plane value of 160cm. The second is a shape-based method, which uses the shape of a group's intersection of ground and head plane projections, and a cost function minimization to approximate the shape of the group. Additionally, their system handles group merging and splitting. Experiments were performed in real-time, on a Pentium IV 3.0 GHz PC, differing camera heights and tilt angles, within distinct locations and illumination conditions. During evaluation, it was noticed that they couldn't get accurate measures in a region far away from the camera, a problem that can be suppressed by using a ROI approach, which neglects any group in a region beyond a certain limit of distance. Both methods show fairly accurate count results, while the shape-based estimator is slightly more accurate than the heuristic-based one. On group estimation, heuristic count obtained an average error of 11.9% while shape count obtained 10.9%. On group size estimation, heuristic count obtained an average error of 15.1% while shape count obtained 11.2%. However, the heuristic-based method is less computationally expensive, achieving higher processing speeds.



**Figure 2.4** – Flowchart of the whole system from [11]. Source: extracted from [11].

Privacy is a common issue in vision systems involving humans, so Chan et al. [20] present a privacy-preserving system to estimate the size of crowds, as illustrated in Figure 2.5. This system doesn't depend on object detection or feature tracking, and also it does not produce a visual record of the people in the scene. Instead, they adopt the mixture of dynamic textures [21] to segment the crowds moving in different directions. Before extracting features from the segmented regions, they consider the effects of perspective, by linearly interpolating between the two extremes of the scene. They then proceed to extract segment features such as *area*, *perimeter*, *perimeter edge orientation* and *perimeter-area ratio*, as well as internal edge features such as *total edge pixels*, *edge orientation* and *Minkowski dimension*, and also texture features, like *homogeneity*, *energy* and *entropy*. In order to achieve the number of people per segment they use feature vector regression based on a Gaussian process. In the experiments they used a dataset that contains a total of 49,885 pedestrian instances. First, they trained the system with 800 training frames, and then tested on the remaining 1200 frames. The count results are within 3 people deviation from the ground-truth 91% of the time for crowds moving away

from the camera, and within 2 people 98% of the time for crowds moving towards the camera. Additionally, they demonstrate the importance of using multiple different feature subsets for improving the performance of the system.



**Figure 2.5** – Crowd counting system: the scene is segmented into crowds with different motions. Normalized features that account for perspective are extracted for each segment and the crowd count for each segment is estimated with a Gaussian process. Source: extracted from [20].

Ryan et al [3] propose a novel scene-invariance crowd counting algorithm that uses local features to monitor crowd size. This work distinguishes itself from the others because they scale the solution to different environments turning it scene invariant. Using camera calibration, they allow the system to be trained on one or more viewpoint and then deployed on any number of new cameras for testing without further training. They use a foreground segmentation technique proposed by Denman [22] which operates in the YCbCr 4:2:2 colour space and provides some invariance to lighting changes. Before extracting features from segmented regions, they use camera calibration to compensate for changes in camera position. Local features like *area*, *perimeter*, and HOG are computed in order to estimate the number of people in the group. Finally, they adopt a Gaussian Process regression to infer the crowd density. This work presents results with several crowd counting datasets such as UCSD, PETS 2006/2009 and QUT.

## 2.3 Individual pedestrian detection

In this approach, each individual is detected separately from the input images and the algorithm estimates the number of people based on the number of detected pedestrians. Individual detection and classification can be achieved by boosting appearance and motion features, Bayesian model-based segmentation, or integrated top-down and bottom-up processing. With the need to detect whole pedestrians, these methods tend to lower performance in very crowded scenes with significant occlusion rates. This problem has been addressed to some extent by adopting part-based detectors.

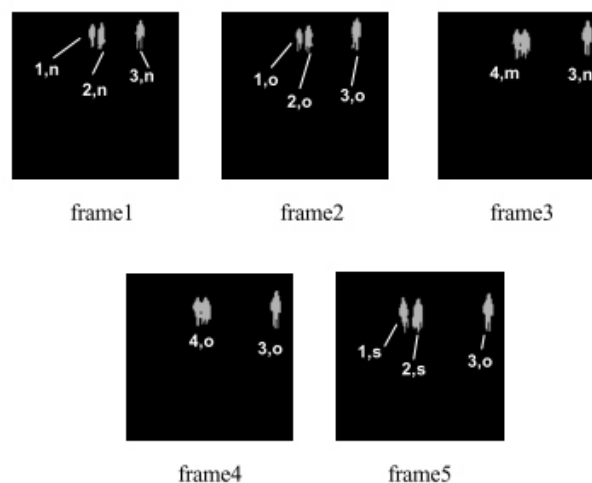
Schofield et al. [23] used yet another approach to handle background subtraction and people segmentation. They perform background segmentation by training RAM based neural networks, resulting in images ready to be processed and analyzed. This method only applies to people detection and background segmentation on single specific images. Tracking or counting people on a sequence of frames was not considered. However, this method enables the algorithm to deal with varying lighting conditions.

Haritaoglu and Flickner [24] developed a system to determine shopping groups in stores. For background segmentation they used a background subtraction model that utilizes color and pixel intensity values over time, in order to classify pixels as foreground, background or shadow. Foreground pixels are then segmented into individual blobs representing people, using temporal and global motion constraints. These individuals are

tracked over-time using a model based on color and edge densities. Experimentation was focused on determining how many individuals make a shopping group and detecting these groups. Similarly to other systems that involve time to help on segmentation, if groups move on the same direction with the same speed, the system's algorithm wouldn't perform well.

Hashimoto et al. [25] designed their own specific imaging system for their people counting system. Using IR sensitive ceramics, mechanical chopping parts and IR-transparent lenses, they developed a highly accurate system that could count passers at a 95% precision rate. They use background subtraction to create thermal images that are then processed to achieve the people counts. The downsides of this system are mainly the lower performance with larger crowds and the strict overhead camera position. Additionally, this approach doesn't handle the occlusion problem as it needs at least 10cm between individuals to count them properly. Within public urban scenes, where crowd counting and occlusion are fairly common, this system would lead to high error counting rates.

Tesei et al. [26] use image segmentation and "long-memory" to track people while handling occlusion. They use background subtraction to achieve areas of interest, followed by thresholding the result to highlight these objects of interest. Using feature extraction on the resulting blobs, such as area, perimeter, bounding box, height, width and centroid, the systems tracks them from frame to frame, while keeping track of all the tracking information. By storing tracking data for each blob, this system manages to handle occlusion fairly well. When two individuals, each one with a blob assigned to a label, occlude themselves, their corresponding blobs result in one single instance. However, when they become separated again, the system manages to label each person correctly to their original labels, by using stored tracking information, as illustrated in Figure 2.6. This system loses efficiency as the number of people increases. Additionally, if the occlusion lasts until the person leaves the field of view, the system couldn't handle this situation, resulting in error counting.



**Figure 2.6** – A sequence of images showing critical cases of blob splitting, merging and displacement. Source: extracted from [26].

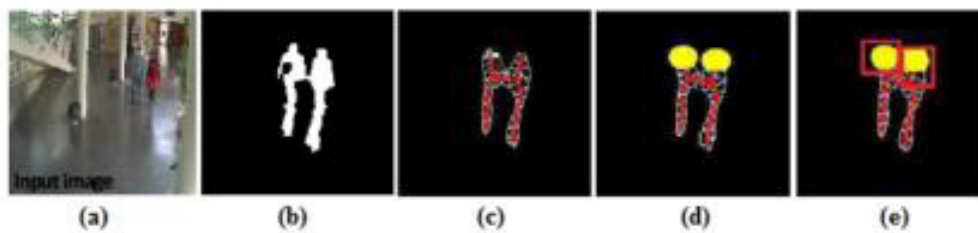
Shio and Sklanksy [27] use extra cues to simulate the perceptual grouping that occurs in human vision, in order to improve on background segmentation algorithms. First, it calculates motion estimations from consecutive frames to determine the boundaries between people and to improve people segmentation. They noticed that, over a few seconds of time, all parts of the same person move as a group in the same direction, even if these parts move in different directions between consecutive frames. This leads to the actual segmentation, which uses a probabilistic model to segment individual people in a moving frame sequence. However, this system would likely fail or at least not perform well when the scene involves a large group of

people all moving in the same direction at the same speed. The paper proves that using extra information such as probabilistic object model can improve segmentation and provides a possible way to handle occlusion.

Conrad and Johnsonbaugh [28] use again the overhead camera position in attempt to simplify the process of segmentation and counting. Instead of using background subtraction they use frame differentiation, which handles much better illumination changes in the scene. This system considers only a small window of the full scene, perpendicular to people's motion flow. They make assumptions of minimum and maximum width for a person and the amount of noise in their images to determine the number of people present on the window, at a given time. To determine direction of movement, they consider the position of each person's center of mass through consequent frames. This system, with simplified image processing algorithms, achieved highly accurate results with a 95.6% performance rate with a sample of over 7491 people. However, it again uses the strict overhead camera position and the performance would surely drop with higher people traffic.

## 2.4 Other approaches

Aziz et al. [13] developed a method that uses skeleton graphs for people counting in crowded environments, as illustrated in Figure 2.7. An input image is segmented into blobs of moving objects, using a forward-background approach developed by Ge et al. [29] For each detected region (individual/group) they compute each graph skeleton using the method developed by Thome et al. [30], achieving fast and accurate results. The skeleton points are then classified on their neighborhood degree. For head detection they consider the points' set having a single neighbor, the segment corresponding to the extreme node is subsequently taken and its inclination degree is compared to the vertical axis, in order to determine if it can be classified as a head of a person or not. To validate each detected head, they estimate the distance between the local reference model of a head in the world coordinate system and a reference detection in the camera coordinate system. In order to reduce the error due to occlusions they finally proceed to head tracking, adopting a framework based particle filter. The experiments were conducted using different videos from CAVIAR dataset. The results were very promising, as they achieved counting accuracies ranging from 75% to 100%, even with crowded and highly human groupings scenes, which have inherently higher occlusion and noise values.



**Figure 2.7** – The counting people system from [13]. (a) Input image, (b) Background subtraction, (c) Skeleton graph, (d) Head detection and pose estimation, (e) Head tracking. Source: extracted from [13].

Idrees et al. [1] try to solve people counting in extremely dense crowd images. The main difference from the other approaches is the crowd density on which they tested their system, containing between 94 and 4543 people per image, with an average of 1280 people over fifty images in the dataset. The proposed framework starts by counting individuals in small patches uniformly sampled over the images. Given a patch  $P$ , they estimate the counts from three complementary sources, which are later combined to obtain a single count estimate for the patch. Images of dense crowds reveal that the bodies are almost entirely occluded, therefore, they rely on HOG based head detections for the first count estimation. Secondly, they noticed that a massive crowd is inherently repetitive in nature, giving the opportunity to capture the crowd density by Fourier



Transform. Additionally they use interest points not only to estimate counts but also to get a confidence whether the patch represents a crowd or not. Finally, different fusion patches are placed into a MRF framework with grid structure, in order to smooth the counts. This paper demonstrates a different approach to people counting in extremely dense crowds, on a scale not tackled before. They achieved errors as minimum as 0% for an image with 426 individuals and 61% for an image with 3333 individuals, claiming that the algorithm performs better for middle range samples, between 1000-2500 individuals per image.

## 2.5 Summary

In this chapter were described the most relevant works from the studied literature. After getting acquainted with various methods for counting people it is now possible to evaluate their major advantages and disadvantages and contextualize them considering the proposed dissertation.

The existing work on counting people with computer vision is relatively recent. As such, there are still several issues that need to be addressed. More than that, there are few works that specifically focus on computational resources and memory usage.

Systems that use stereovision [7]–[9], [31] can perform with better overall accuracy. However, this increases system's cost, as two cameras are required instead of one and the computational demand gets higher which may require fast processing units to run the algorithms.

The trajectory cluster approach, seen in [7], [8], [15], [16], extracts features over time and aggregates them for eventual analysis. Feature data must be recorded sequentially, which may require higher memory capacity. Furthermore, this method relies on people movement dynamics to maintain relatively predictable. This can be applicable to some counting environments, like an entrance of a subway station or a corridor, but can lead to high error results in outdoor urban environments, where crowds move in different directions at variable speeds.

Individual pedestrian detection, seen in [23]–[28], requires single segmentation of each person. This leads to an overall better classification of human individuals and the system should be less vulnerable to non-human objects like bicycles or animals. However, this involves full human detection, which brings up privacy issues that should be avoided in this dissertation. Additionally, this method need precise individual segmentation which is difficult to secure under urban environments, where occlusion and crowd densities are higher.

Feature-based regression, seen in [10]–[12], [14], [32]–[35], is a method with large versatility that has recently been adopted in various urban systems for counting people. The computational cost of a system based on this approach depends on different factors, such as the number of extracted features and the chosen regression model. It can preserve privacy and doesn't rely on individual detection to achieve accurate results. Additionally, the camera position is not very restrictive and many authors tested their systems with varying tilt angles.



## Chapter 3

# Image descriptors for people counting

This chapter presents a perspective invariant approach for people counting in urban scenarios. Perspective invariance can be described as counting the same number of people regardless of their relative position to the camera point of view. This differs from scene invariance, where the camera can change position, tilt or even the capture environment. The presented method uses local features to estimate the number of people within each individual group of foreground pixels. The resulting sum of estimates returns the final count for a given frame. This approach offers some advantages against the largely used (e.g. [32], [36], [37]) holistic systems, which extract holistic features of the scene in order to evaluate its crowdedness level. Using local features requires less training frames to achieve reliable estimates, as opposed to holistic features, which may need hundreds [12] or even thousands [37] of frames, due to the wide inconsistency in crowd behaviours, distribution and overall size. Additionally, the count estimation can be done for a specific region of the capture scene, unlike holistic approaches that can only provide a density for the whole scene.

In order to evaluate image descriptors for people counting, it was required to design and construct a complete pipeline system to infer the crowd density within a scene, using said image descriptors. The regression model must be capable of estimating the number of people given a specific set of calculated features. In order to do this, the system must be previously trained with ground truth annotated instances. The proposed pipeline can be summarized into the following sequence of steps:

1. Dataset selection
2. Background subtraction
3. Perspective normalization
4. Ground truth annotation
5. Feature extraction
6. Regression models

Although the main focus of this thesis consists of feature evaluation and optimization for people counting, this chapter presents all the methods utilized that lead to final results. The remainder of this chapter is structured as follows: Section 3.1 discusses the dataset selection stage; Section 3.2 describes the background subtraction algorithm and the postprocessing routine; Section 3.3 explains the perspective normalization and how this relates to scene invariance; Section 3.4 presents the ground truth annotation method utilized; Section 3.5 details the feature extraction process; Section 3.6 proposes an extension to a powerful image descriptor; Section 3.7 presents the chosen regression models; Section 3.8 explains the utilized measurement metrics; and Section 3.9 presents the summary for this chapter.

### 3.1 Dataset selection

The starting point of the proposed system consisted of dataset selection for people counting purposes. A dataset provides video or image data with additional annotations, in this case the number of people, which can be used as workbench for computer vision applications. Analyzing the existing literature on the theme, 8 datasets were considered (Table 3.1).

**Table 3.1** – Nine dataset were considered to develop the proposed crowd counting algorithm. A subset of the total number frames was annotated at constant frame spacing indicated by the column interval. FC dataset was constructed on a later development stage of the thesis.

Descriptor	# Frames	# Annotated	Interval	Max crowd
PETS 2009, View 1	460	46	10	32
PETS 2009, View 2	460	46	10	32
PETS 2006, View 3	3000	120	25	5
PETS 2006, View 4	3000	120	25	6
QUT, Camera A	10400	50	200	8
QUT, Camera B	5300	50	100	23
QUT, Camera C	5300	50	100	10
UCSD	2000	2000	1	45
FC	408	408	1	19

As the focus of this system is crowd counting in urban outdoor environments the 5 datasets of PETS 2006 [38] and QUT [39] were excluded. This leaves both views from PETS 2009 dataset and UCSD dataset. The UCSD crowd counting dataset [40] consists of 2000 fully annotated frames and provides the maximum crowd density of all the considered datasets. This was a valuable asset that lead to its choice, as it was preferable to have a high number of annotated frames on the dataset, in order to evaluate image descriptors performance with different training and test sets, and PETS 2009 [41] only provided 46 annotated frames for each view, which narrows the possibilities for this type of experimentation.

Later on, a newly made dataset from the Future Cities team, denoted FC, was made accessible for the development of this thesis. This dataset was also captured on an outdoor urban environment, with a maximum crowd density of 19 people and provided 408 fully annotated frames. The annotation process is further detailed in Section 3.4. FC dataset was used on later experiments, when the counting algorithm was already built, in order to evaluate its performance on a different scene.

### 3.2 Background Subtraction

For this counting approach it is considered a static camera that retrieves visual data of the capture scene over time which, in this case, was already assembled into UCSD dataset as a sequence of frames. Once the capture subsystem is established, it is essential to detect people in each frame, which can be done using the so called background subtraction. This process is based on the assumption that the images of the scene without disturbances show some regular behavior that can be described by a statistical model. If this statistical model of the scene is achieved, an intruding object can be detected by finding the areas of the image that don't fit the statistical model.

The foreground mask is obtained using a background subtraction algorithm proposed by Z.Zivkovic in [42], which improves the Gaussian Mixture Model (GMM) proposed in [43]. This algorithm constantly updates the parameters of the GMM and the number of components for each pixel in an on-line procedure, resulting in an auto adaptation to the capture scene in real-time. This is useful for the proposed counting system as it is focused on outdoor environments, where daytime or weather conditions can cause gradual changes in the illumination of the capture scene. The background subtraction algorithm is available in the OpenCV method MOG2 [44].

The input images used for background subtraction use 8-bit grayscale, which is composed solely of varying pixel intensities, where black is the weakest intensity (0) and white is the strongest (255). Using this pixel intensity range, the processing time of the algorithm is reduced, when compared with colour spaces like RGB, YCbCr or HSV. UCSD dataset videos have 238 x 158 resolution at 10 fps. FC dataset video has 320 x 240 resolution, at 1 fps.

The output masks from the background subtraction algorithm, for both UCSD and FC datasets, were provided by thesis' supervisor and the rest of the involved team from the Future Cities Project. The parameters used are detailed on the following table.

**Table 3.2** – Parameter values used in MOG2 method, for both UCSD and FC dataset.

Parameter		Dataset	
type	name	UCSD	FC
int	nMixtures	2	5
bool	detectShadowsMOG2	false	false
int	historyMOG2	1000	1000
float	thresholdMOG2	4.0f*8.0f	4.0f*8.0f
float	varThresholdGen	3.0f*3.0f	3.0f*3.0f
float	backgroundRatio	90.0	90.0

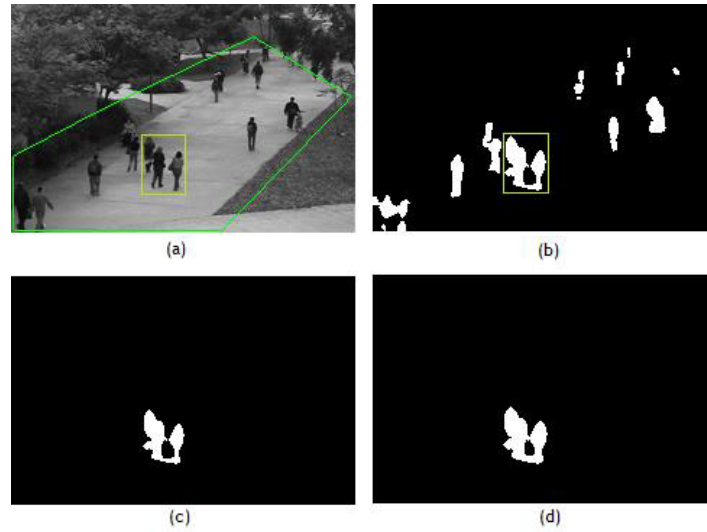
The foreground masks determined by Zivkovic algorithm are tested to determine which 1-valued pixels are contained in the desired ROI (Figure 3.1). Those are stored as binary images with 0 and 1 pixel intensity values, where groups of 1-pixels represent an object that doesn't belong to the background of the capture scene. These objects may or may not be people so it is necessary to determine whether they should be accounted as such.

### 3.2.1 Postprocessing

As the proposed counting algorithm is based on local features of foreground regions, it is necessary to divide obtained foreground masks into individual submasks. Individual blobs from the original foreground mask are labeled considering connected pixels with 8-connectivity and each labeled region is subsequently stored as a new mask.

It was observed that many of the segmented masks did not contain all the expected pixels from its corresponding grayscale image. This was not desirable as it would crop groups of people and could lose important pixel values for further feature extraction. To avoid this situation, following foreground segmentation, a morphological dilate operation is applied to each binary mask in order to obtain a slightly larger binary mask (Figure 3.1). This method is not optimal as it may introduce pixels that belong to background or overlap binary masks doubling the counting region. However, using a flat, diamond-shaped, 1 pixel radius

structuring element for dilate operation, this negative effect was minimized. Further experimentation confirmed that this postprocessing procedure had positive impact on overall system's accuracy.



**Figure 3.1** – Background Subtraction algorithm. (a) Example frame of the UCSD dataset. The ROI for this scene is marked by the green line. Yellow box marks the considered blob for (c) and (d). (b) Corresponding foreground mask with ROI obtained by the BS algorithm. (c) Foreground submask for one segmented blob. (d) The same blob after dilate morphological operation.

Some of the descriptors discussed in Section 3.5 are calculated directly on binary masks such as the one illustrated in Figure 3.1(d). Other descriptors need full grayscale images to extract and calculate its corresponding features. To generate these grayscale images it was determined the Bounding Box for each connected submask and subsequently cropped the original grayscale frame into a smaller image limited by the Bounding Box coordinates (Figure 3.2).



**Figure 3.2** – Example of images for feature extraction. (a) Foreground binary mask. Yellow line marks blob Bounding Box. (b) Grayscale image for the corresponding blob of (a).

### 3.3 Perspective normalization

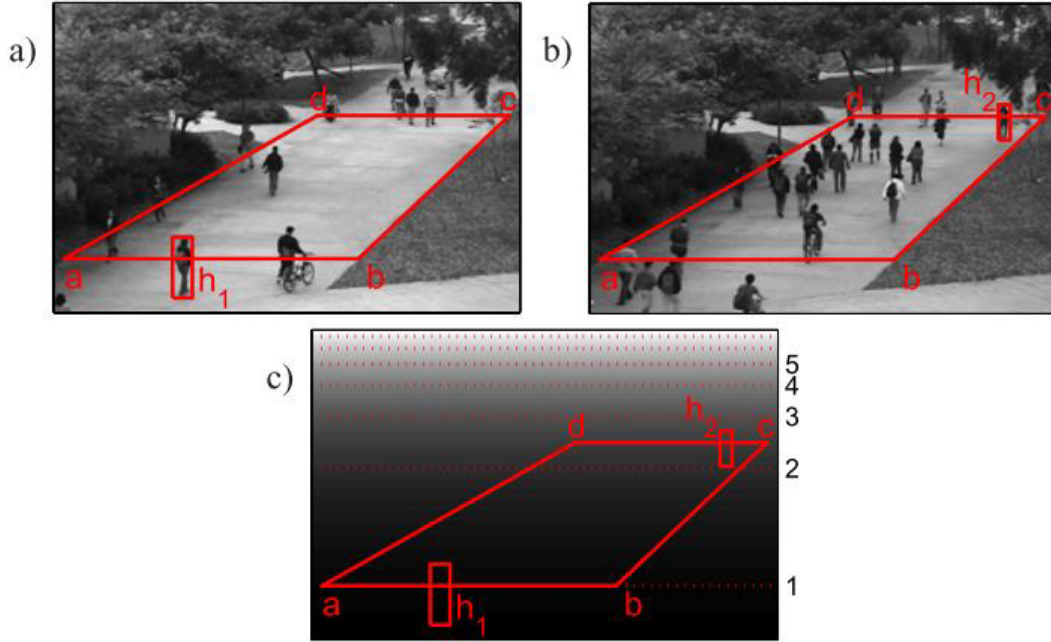
Before extracting features from the foreground regions it is important to consider perspective effects as well as camera distortion. Although the proposed system is designed to operate over a single viewpoint and a static capture scene, perspective issues are also important to take into account for systems with multiple viewpoints and variable scenes.

Objects that are closer to the camera appear larger than more distant objects. Consequently, features extracted from closer objects would account for a larger portion of the foreground mask than ones extracted from an object that is farther away. Therefore, it is important that extracted features are normalized

appropriately so that the trained algorithm can effectively count people, independently of their relative position to the camera.

The method that was used for perspective normalization was the one proposed by Chan in [14]. This approach uses a perspective normalization map to weight each foreground pixel, with larger weights given to farther objects. In order to calculate the perspective map, they linearly interpolate the two extremes of the scene, following the sequence of procedures:

1. A ground plane is marked in the scene (Figure 3.3a) and the distances  $|\overline{ab}|$  and  $|\overline{cd}|$  are measured;
2. A reference pedestrian is selected and the heights  $h_1$  and  $h_2$  are measured when the center of the person is on  $|\overline{ab}|$  (Figure 3.3a) and on  $|\overline{cd}|$  (Figure 3.3b);
3. The pixels on  $|\overline{ab}|$  are given a weight of 1, and the pixels on  $|\overline{cd}|$  a weight of  $\frac{h_1|\overline{ab}|}{h_2|\overline{cd}|}$ ;
4. The remaining pixels weights are calculated by linear interpolation between the two lines (Figure 3.3c).



**Figure 3.3** – Perspective map for the UCSD dataset. a) Reference person at the closer extreme of the scene, and (b) at the distant extreme. b) The perspective map which scales pixels by their relative size in the true 3D scene. Source: extracted from [14].

Figure 3.3c illustrates the perspective map of this scene, obtained by following the above steps. The perspective map will be denoted as  $D_2$ . For 2 dimension features, such as area, the weights of the map are directly applied to each pixel, while for 1 dimension features such as perimeter and edges, each pixel is weighted by the square-root of the original map, denoted as  $D_1$ :

$$D_1(i, j) = \sqrt{D_2(i, j)} \quad (3.1)$$

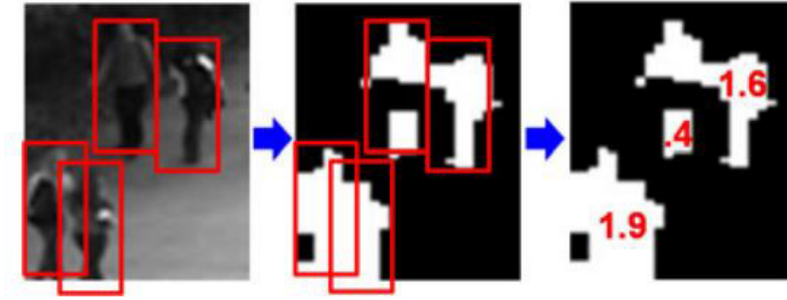
This method provides a simple solution to account perspective effects on feature extraction, even when camera calibration is not available. However, it does not compensate different camera positions or changeable capture scenes. Because this approach utilizes reference pixels instead of real world reference, it is not appropriate for scene invariant people counting. To surpass this issue, the system must be trained and tested on the same scene and viewpoint, which would be a negative aspect if it was desired to scale the system to other environments.

### 3.4 Ground truth annotation

Because the system computes local features of each foreground blob obtained by the background subtraction algorithm, training must also be performed on a local level. This requires ground truth annotation to specify a people count for each segmented blob. However, as foreground segmentation is not perfect, some blobs are prone to errors such as splitting, fading and noise, which makes the annotation process more complex, when attempting to assign fractional counts.

The approach that was used for local ground truth annotation follows a method referred to as ‘dotting’ by Lempitsky [35], because it only needs the user to click on the center of each object in the image in a GUI. The surrounding region of an individual is then estimated by the contour of a rectangle model. Each side of this model is divided by the density map  $D_I$ , determined in the previous section, in order to adjust the rectangle to the individual’s dimension, regardless of their relative position to the camera.

Once ‘dot’ annotations for the desired objects are done, the algorithm performs blob annotations automatically, by assigning the annotated individuals to their confined foreground blobs. This is done by overlapping foreground blobs and the rectangle bounding regions. This process ensures that fragmented objects of the same person are assigned to the same individual ‘dot’ annotation. On the other hand, if multiple persons result in a single blob, their corresponding bounding regions will overlap this blob (Figure 3.4).



**Figure 3.4** – Ground truth annotation process. Manual annotations (left) are overlayed on the foreground segmented objects (centre), and the region overlaps are used to automatically determine ground truth counts for each blob (right). Source: adapted from [45].

Local blob counts are achieved using set notation. Considering the defined regions from Table 3.2, the following values are calculated [45]:

- $Q_i$ : the ‘amount’ of person  $i$  within the scene’s ROI:

$$Q_i = \frac{|M \cap R_i|}{|R_i|} \quad (3.2)$$

- $C_{in}$ : the ‘contribution’ of person  $i$  to blob  $n$ :

$$C_{in} = \frac{|R_i \cap B_n|}{|R_i \cap B|} \times Q_i \quad (3.3)$$

- $f_n$ : the total number of people represented by blob  $n$ :

$$f_n = \sum_i C_{in} \quad (3.4)$$



**Table 3.3** – Considered regions treated as sets of pixels using set notation.

Notation	Description
$M$	Mask of the scene (ROI).
$F$	Foreground pixels detected using an adaptive background subtraction algorithm [42].
$B$	Foreground pixels within ROI mask, i.e. $B=M \cap F$ consists of blobs $\{B_n\}$ .
$B_n$	Blob $n$ within $B$ , where $B=\cup_n B_n$ .
$R_i$	Bounding region of person $i$ . (This may be inside the ROI, partially inside at the edge, or outside.)
$R_i \cap B_n$	The foreground pixels inside $R_i$ belonging to blob $B_n$ .
$R_i \cap B$	The foreground pixels inside $R_i$ .

Therefore,  $f_n$  value gives the total number of people represented by blob  $n$ , which is the local target count desirable to train the system. This process is calculated independently from the foreground segmentation stage, simplifying the annotation process. This method also has the advantage to allow some tolerance for errors in the background subtraction stage, as it assigns zero value count annotations to small blobs generated by noise.

The holistic ground truth or, in other words, the total annotated count for the whole scene, is measured by considering the number of pedestrians whose manual ‘dot’ annotations lie within the ROI, summing local annotations for each blob.

### 3.5 Feature extraction

In order to estimate the number of people present in each segmented blob, it is necessary to compute various features that describe their respective image segments. Several image descriptors were used to calculate different features at a local level, rather the holistic level of the scene. Using statistics given by image descriptors, the system can be trained in order to count people in the segmented foreground. However, not all the presented descriptors were extended for people count estimation. Instead, a ranking algorithm was performed using typical parameters for each descriptor, in order to infer the power of their composing features for the proposed algorithm. The feature selection stage was based on the output of the ranking algorithm as it is described in Section 4.1.

Most of the utilized image descriptors were adapted from previous work of thesis’ supervisor D.Moura in [46]. This study presents an evaluation of image descriptors combined with clinical data for breast cancer diagnosis. These descriptors were used at low-level, to extract local features from masses and calcifications, so it was decided to adapt them to this thesis, as the images are also analyzed locally. From this work, 1 novel and 10 conventional descriptors were calculated for the obtained foreground regions, in order to evaluate which ones could be viable for people counting purposes.

In addition, features commonly used for people counting purposes were also computed, including ones used by Chan [32] and Ryan [45].

### 3.5.1 Intensity descriptors

#### *Intensity statistics*

This descriptor calculates statistics over the gray levels of the pixels belonging to foreground patch. These features include mean, standard deviation, skewness, kurtosis, minimum intensity and maximum intensity, making a total of six features.

#### *Histogram measures*

This descriptor calculates statistics over the gray level histogram of the foreground patch. Extracted features include six statistical measures [47]: average intensity, contrast, smoothness, skewness, uniformity and entropy.

#### *Invariant moments*

This set of seven features is calculated using Hu's approach [48]. These features are based on statistical moments that are invariant to translation, scale, and orientation of the observation.

#### *Zernike moments*

This descriptor uses Zernike moments [49], which are constructed using a set of complex polynomials that describe a unitary disc (radius = 1). The descriptor defines a circular patch by the coefficients of the polynomials. The first polynomial (order 0) has only one term with coefficient equal to the average pixel intensity. In contrast to statistical moments and invariant moments, Zernike moments have an orthogonal basis, which guarantees independent coefficients, and they also remain invariant to translation, rotation, and scale.

### 3.5.2 Texture descriptors

#### *Haralick features*

Haralick features [50], are calculated from the gray level co-occurrence matrix (GLCM), which is a 2D histogram that measures the 2nd-order joint conditional probability of two grey levels occurring at a given distance  $d$  and at a given direction  $\theta$ . The image is quantized into  $B$  gray levels,  $\theta$  is typically  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  and  $d$  a city block distance  $\geq 1$  pixel. From this matrix, a set of 14 features is computed. Haralick et al. proposed computing these features for the four directions and averaging the results in order to achieve some invariance to rotation. Some studies have included Haralick features for estimating the number of people (e.g.[11], [14]).

#### *GLRL*

Gray level run length (GLRL) analysis [51] calculates the occurrence of sets of consecutive collinear pixels with given length  $l$  and direction  $\theta$  for a given gray level. Gray levels are quantized in  $B$  bins and GLRL matrices are computed for four directions ( $\theta$  is  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ ). For each direction, a set of 11 features is calculated, resulting in a total of 44 features.

#### *GLDM*

Gray level difference matrix (GLDM) stores the occurrence of absolute differences between pairs of gray levels separated by a given distance  $d$  and a given direction  $\theta$ , with the element GLDM being the number of times the grey-level difference is observed at a distance  $d$ . Gray levels are quantized in  $B$  bins and GLDM

matrices are computed for four directions ( $\theta$  is  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ ). For each matrix, a set of 5 features (mean, contrast, entropy, angular second moment and inverse second moment) is calculated, resulting in a total of 20 features. GLDM has been used for estimation of crowd density by Marana et. al in [52].

### 3.5.3 Multi-scale texture descriptors

#### *Gabor filter banks*

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. These filters are frequently used for edge detection, as they detect edges according to filter's orientation and frequency. Furthermore, by adjusting the standard deviation of the Gaussian envelope, it is possible to adjust the degree of blurring. For a given set of orientation, frequency and envelope values, the following features are calculated with the Gabor filter: mean, standard deviation, energy and entropy.

#### *Wavelets*

In signal theory, a discrete wavelet transform enables decomposition of a discrete signals in two sets of coefficients: approximation and detail [47]. Regarding 2D discrete wavelet transform, the decomposition originates an approximation image and three detail images (horizontal, vertical and diagonal), all with half the width and height of the original image. In order to compute features of the coefficients of each level, it is required to define the filters that define the wavelet and the number of levels of decomposition. The same features calculated on Gabor filters were computed for each sub-image that originated from the wavelet transform. Wavelet transforms have been used in people detection and counting studies (e.g. [30], [53]).

### 3.5.4 Shape and size descriptors

#### *Segment features*

These features describe object's shape and size. A set of 3 features is calculated: *Area*, which is the total number of pixels in the blob weighted by the 2D perspective map described in Section 3.3 ( $D_2$ ); *Perimeter*, which is the total number of pixels on the blob's contour weighted by the 1D perspective map described in Section 3.3 ( $D_1$ ); and *Perimeter-area ratio* or *circularity* obtained by  $\frac{4\pi \times Area}{Perimeter^2}$ .

#### *Internal edge features*

A Canny edge detector is applied to the original image and the resulting image is masked by the foreground blob. From the obtained segment, the following features are computed: *Edge length*, which is the total number of edge pixels contained in the segment; *Minkowski dimension*, which is the Minkowski fractal dimension of the edges in the segment, which estimates their degree of "space-filling" [54]. These features are used for privacy preserving crowd counting in [14].

### 3.5.5 Spatial distribution of the gradient

#### *Histograms of oriented gradient*

Histograms of oriented gradients (HOG) describe images through the distribution of the gradient [55]. Images are divided into a grid of blocks and each blob is described by a histogram of the gradient's orientation. Each histogram is constructed according to a given number of orientation bins that divides the range of possible orientation (from 0 to  $2\pi$  radians). The value of each orientation bin is calculated by summing the magnitude

of the gradient of pixel that have gradient direction within the limits defined by the orientation bin. Additionally, histograms can be normalized, with the most common normalizations being the L1 and L2 norm [52]. HOG and HOG based descriptors have been commonly used for people detection and crowd counting algorithms (e.g. [1], [3], [55], [56]).

### *Histograms of gradient divergence*

Histograms of gradient divergence [46] is a rotation invariant image descriptor that measures shape regularity. Assuming that the object is centered on the patch, gradient divergence of a pixel  $P$  is measured as the angle between the vector of the intensity gradient on  $P$  and a vector with origin on  $P$  pointing to the center of the patch. To account for divergence of the gradient, HGD also considers the distance of the pixel to the center using  $R$  regions, with each region being described by a histogram with  $B$  orientation bins. Rotation invariance is achieved by using concentric regions and by storing the divergence instead of the orientation of the gradient.

### **3.5.6 Spatial autocorrelation**

#### *Moran's I Geary's C*

In statistics, Moran's I is a measure of spatial autocorrelation developed by P. Moran. Spatial autocorrelation is characterized by a correlation in a signal among nearby locations in space. Spatial autocorrelation is more complex than one-dimensional autocorrelation, because it is multi-dimensional and multi-directional. Moran's I is inversely related to Geary's C, but it is not identical. Moran's I is a measure of global spatial autocorrelation, while Geary's C is more sensitive to local spatial autocorrelation. A set of 12 are extracted with this descriptor

## **3.6 Perspective invariant Histograms of Oriented Gradient**

Quoting W. Schwartz [57], "The work of Dalal and Triggs [55] is notable because it was the first paper to report impressive results on human detection. Their work uses HOG as low-level features, which were shown to outperform features such as wavelets [58], PCA-SIFT [59] and shape contexts [2]." Additionally, HOG is an image descriptor that has proven value in several works on people counting. Furthermore, the performed ranking algorithm on the UCSD dataset, placed HOG among the top descriptors for people counting purposes, as it can be seen in Section 4.1. In this thesis is proposed a new image descriptor, HOGp, that extends HOG, in order to make it invariant to perspective.

Perspective invariant histograms of oriented gradient, denoted HOGp, introduces weighted votes in HOG computation. Because HOG is a gradient based descriptor, it can be normalized using the 1D perspective map described in Section 3.3 ( $D_I$ ), in order to normalize the effects of camera perspective across the whole scene. The gradient of a grayscale image  $f$  is given by the formula:

$$\nabla f = \frac{\partial f}{\partial x} \hat{x} + \frac{\partial f}{\partial y} \hat{y} \quad (3.5)$$

where  $\frac{\partial f}{\partial x}$  is the gradient in the  $x$  direction ( $G_x$ ) and  $\frac{\partial f}{\partial y}$  is the gradient in the  $y$  direction ( $G_y$ ).

The gradient direction of pixel  $(i,j)$  can be calculated by the formula:

$$\angle \nabla f(i,j) = \text{atan}\left(\frac{G_y(i,j)}{G_x(i,j)}\right) \quad (3.6)$$

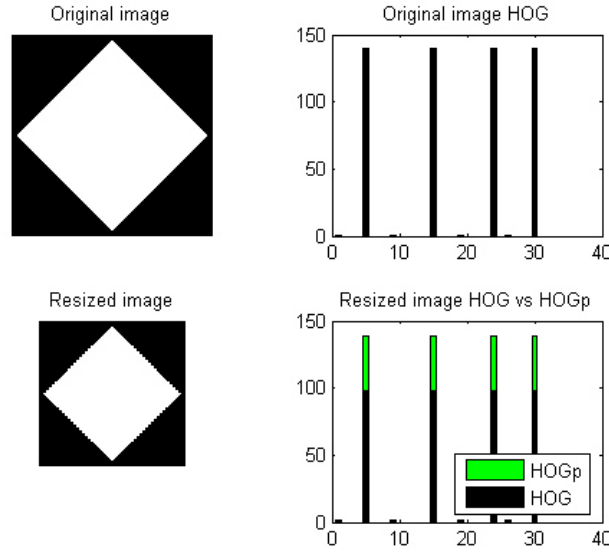
The gradient magnitude of pixel  $(i,j)$  can be calculated by the formula:

$$|\nabla f(i,j)| = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \quad (3.7)$$

The contribution of each pixel  $(i,j)$  to a histogram bin is proportional to the gradient magnitude  $|\nabla f(i,j)|$ , and it is also weighted by the 1D density map  $D_1(i,j)$  to normalize for perspective. Considering that the value of the  $h$ th histogram bin is  $E_n[h]$ , and the orientation angle for that bin is lower bounded by  $\theta_h$ :

$$E_n[h] = \sum_{(i,j)} \begin{cases} D_1(i,j) \times |\nabla f(i,j)| & \text{if } \theta_h \leq \angle \nabla f(i,j) < \theta_{h+1} \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

Figure 3.5 illustrates an example of HOG normalization using HOGp method with an adequate density map. The resized image is half the blob area of the original image, while maintaining shape and texture characteristics, in order to simulate a closer and a distant object, in terms of perspective.



**Figure 3.5** – HOGp simulation. Bottom image has half the blob area of top image. It is observable that, for a perfectly resized image, the flat histogram bins from HOGp match the values for the original image, while HOG does not.

### 3.7 Regression Models

Given a set of features and their target value, the regression builds a model that is used to infer the target value of new instances described by the same features. In this section are described the models used to infer the number of people from feature vectors. To test the proposed system, nine regression models were trained using several training sets from both the UCSD and FC datasets.

### **Linear Regression**

Linear regression is a statistical modeling technique used to describe the relationship between a scalar dependent variable  $y$  and one or more explanatory variables denoted  $X_i$ . The general equation for a linear regression model is:

$$y = \beta_0 + \sum \beta_i X_i + \varepsilon_i \quad (3.9)$$

For the above equation,  $\beta_i$  is a parameter vector, which elements are called effects, or regression coefficients. Statistical estimation and inference in linear regression focuses on  $\beta_i$ .  $\varepsilon_i$ , is called the error term, disturbance term, or noise. This variable captures all other factors which influence the dependent variable  $y$ , other than the  $x_i$ .

The method used to select features for use in the Linear Regression was M5's method, which steps through the features removing the one with the smallest standardized coefficient until no improvement is observed in the estimate of the error. Existing approaches use linear regression for people counting (e.g. [12], [33], [60]).

The Linear Regression was also enhanced by two Meta-algorithm denoted Additive Regression [61] and Bootstrap aggregating [62]. Additive Regression is done by fitting the regression model on each iteration, to the residuals left by the regression on the previous iteration. Prediction is accomplished by adding the predictions of each regression. The number of iterations was set to 50 and no shrinkage was applied.

Bootstrap aggregating, also known as Bagging, is an ensemble Meta-algorithm method that creates separate samples of the training dataset and generates a regression for each sample. The results of these multiple regressions are then combined, in order to improve the robustness and accuracy. Additionally, it also reduces variance and helps to avoid overfitting. The sample set size was set to 70% of the training dataset and the number of iterations was set to 50.

### **REPtrree**

Tree-based regression models are known for their simplicity and efficiency, as the final results for regression can be summarized into a series of logical if-then conditions (tree nodes). Therefore, there is no implicit assumption that the underlying relationships between the predictor variables and the dependent variable are linear, follow some specific non-linear link function, or that they are even monotonic in nature. Regression trees are obtained using a fast divide and conquer greedy algorithm that recursively partitions the given training data into smaller subsets. The use of this algorithm is the cause of the efficiency of these methods. However, it can also lead to poor decisions in lower levels of the tree due to the unreliability of estimates based on small samples of cases.

In particular, REPtrree is a fast regression tree learner which builds a regression tree using information gain as the splitting criterion, and prunes it using reduced-error pruning. Additionally, it considers all the attributes to split on at each node. The parameters used were default for Weka version 3.7. This Regression model was also enhanced by both Additive Regression and Bagging, using the same parameters set for Linear Regression.

### **M5P**

M5P combines a conventional decision tree with the possibility of linear regression functions at the nodes. First, a decision-tree induction algorithm is used to build a tree, but instead of maximizing the information gain at each inner node, a splitting criterion is used that minimizes the intra-subset variation in the class values down each branch. The splitting procedure in M5P stops if the target values of all instances that reach a node vary very slightly, or only a few instances remain. Second, the tree is pruned back from each leaf. When pruning, an inner node is turned into a leaf with a regression plane. Third, to avoid sharp discontinuities between the subtrees a smoothing procedure is applied that combines the leaf model prediction with each node along the

path back to the root, smoothing it at each of these nodes by combining it with the value predicted by the linear model for that node.

The parameters used were default for Weka version 3.7. This Regression model was enhanced by Additive Regression using the same parameters set for Linear Regression. Bagging was neglected as the computational resources were not enough to run it properly.

### ***Decision Stump***

A decision stump is a machine learning model consisting of a one-level decision tree. That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. First, a decision-tree induction algorithm is used to build a tree, but instead of maximizing the information.

Decision stumps are often used as components in machine learning ensemble techniques such as Additive Regression, which was the one used in this work, using the same parameters set for Linear Regression.

## **3.8 Evaluation metrics**

The performance attributes used in the following chapter are expressed by the Mean Absolute Error (MAE) of the count estimate per frame, which means that lower MAE values represent higher accuracy rates and vice-versa. This metric is frequently used on related works, so it was chosen to open the possibility for direct performance comparison. For a number of tested frames  $n$ , the estimated count for frame  $i$  denoted  $Ce_i$ , and the ground truth annotated count for frame  $i$  denoted  $Ct_i$ , MAE is given by:

$$MAE = \frac{\sum_{i=1}^n |Ce_i - Ct_i|}{n} \quad (3.10)$$

## **3.9 Summary**

This chapter presented all the methods used to fulfill the established thesis objectives. It starts by explaining the reasons behind the choice for the two datasets that were used in this work, UCSD and FC (recall Section 3.1). Then, Section 3.2 describes the background subtraction method, which is able to extract pedestrians from the rest of the scene, producing foreground masks that were postprocessed in order to improve final results.

Because the cameras used in this work are not calibrated, Section 3.3 presents a method to introduce perspective effects on the algorithm.

To estimate the number of people per mask, the system needs to be trained with proper annotated instances and a method to do so is explained in Section 3.4.

Image descriptors are methods used to extract statistics and features from images. Descriptors used in this work are described in Section 3.5. Section 3.6 thesis' contribution: a proposal of an extension to a remarkable image descriptor that is vastly used in people detection. The proposed image descriptor was called perspective invariant Histograms of Oriented Gradients (HOGp) and it can be summarized as using density maps to weight image pixels, in order to normalize HOG for perspective.

Once the desired features are calculated, a regression model needs to be trained to estimate the number of people per segment. Section 3.7 describes the different regression models that were used in this work.

Finally, Section 3.8 details the evaluation metric that was chosen to access image descriptors' performance.

By following this series of methods, it was possible to build a system capable of performing people counting experiments on adequate datasets. The implemented algorithms were coded using Matlab, and they

can be easily extended to add more image descriptors, properly annotated datasets and Weka based regression models. Using this pipeline, a series of experiments were carried in order to evaluate and optimize image descriptors for counting people in urban scenarios. The following chapter presents the results for these experiments.



# Chapter 4

## Results

This chapter presents experimental results of the proposed algorithms and implemented methods. Experiments were conducted using two datasets: the UCSD Crowd Counting Dataset, which has 2000 fully annotated frames, and FC Dataset which has 408 fully annotated frames.

The performance attributes used in this chapter are expressed by the Mean Absolute Error (MAE) of the estimated number of people per frame, which means that lower MAE values represent higher accuracy rates and vice-versa.

Regarding software, the following experiments were carried using: OpenCV Version 2.4.8 for background subtraction; Matlab Version R2013b for image post-processing and feature extraction; and Weka Version 3.7 for feature selection and regression.

Regarding computational resources, the following experiments were conducted on a PC with the following specifications: Windows 8.1 Pro (64-bit) OS; Intel(R) Core(RM) i7-2630QM CPU @ 2.00 GHz 2.00 GHz processor; and 4.00 GB RAM

Section 4.1 presents results for the image descriptor ranking algorithm. Section 4.2 shows results for sensitivity analysis of the descriptors' parameters. Section 4.3 presents crowd counting results for the full UCSD Dataset, along with accuracy comparison of the chosen Regression Models. Section 4.4 demonstrates the impact of the training set size. Section 4.5 presents final experimental results on FC dataset. In Section 4.6 is made a brief summary and discussion of the results.

### 4.1 Ranking of image descriptors

In order to estimate the inferring power of image descriptors for urban crowd counting purposes it was used an attribute ranking algorithm, adopting all the descriptors described in Section 3.5 as input. The chosen feature selection algorithm was Relief-F proposed by Kira and Rendell [63]. It evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. The input consists of 15 image descriptors with varied number of features, for a total of 458 features. Although most feature ranks (Table 4.1) are grouped by their respective descriptor, segment features (area, perimeter and circularity) and internal edge features (edge length, Minkowski dimension) were not grouped, in to order to evidence their individual power, as they are used separately in some previous works on people counting.

**Table 4.1** – Relief-F feature selection algorithm output for the UCSD Crowd Counting Dataset. Ranking results for 15 image descriptors with different number of features, for a total of 458 features. Feature attributes are grouped by their respective image descriptor and ordered by best to worst score.

Descriptor	Number of Features	Average score	Best score	Average rank	Best rank
Zernike Moments	66	0.0282623	0.0376845	55	1
HOG	72	0.0218048	0.0349830	107	6
HGDn	16	0.0116201	0.0318668	240	20
Moran Geary	12	0.0084585	0.0301119	277	32
Minskowski	9	0.0123584	0.0300319	210	35
Perimeter	1	0.0255154	0.0255154	80	80
Wavelet	52	0.0043796	0.0204948	327	111
HGD	16	0.0089171	0.0196184	264	119
HOGn	72	0.0058322	0.0193435	304	120
SimpleGL	6	0.0053875	0.0192132	305	121
GLRL	88	0.0069092	0.0190871	282	123
Gabor	4	0.0056777	0.0105251	301	215
Edge Length	1	0.0095432	0.0095432	224	224
Haralick	13	-0.0104002	0.0094669	387	225
Area	1	0.0070508	0.0070508	254	254
GLDM	20	0.0055575	0.0068671	290	255
Invariant Moments	7	0.0029855	0.0055516	326	294
Circularity	1	0.0023978	0.0023978	365	365

The achieved ranking results (Table 4.1) give some proof that Zernike moments and Histograms of Oriented Gradients (HOG) can be powerful descriptors for crowd counting algorithms. If the average rank of each descriptor is considered, it is observable that blob perimeter is placed higher on ranking results, losing only to Zernike moments. Nonetheless, blob area and circularity achieved a lower score than expected, losing to several other descriptors.

## 4.2 Sensitivity analysis of the descriptors' parameters

This section presents the analysis of the image descriptors' parameters and how they influence the counting algorithm accuracy. All results shown in this section were achieved using the original training set of the UCSD Dataset (frames 601:1:1400), with the first 66% frames as train set and the remaining 33% as test set. These results are all relative to Linear Regression model.

The evaluated image descriptors were: Histogram of Oriented Gradients (HOG) and Zernike moments (Zer). Furthermore, at this stage of progress it was proposed a new descriptor based on HOG, which was called and referenced as HOGp, with *p* standing for *perspective* (recall Section 3.6). In order to evaluate HOGp efficiency, it is directly compared with its precursor descriptor – HOG.

### 4.2.1 HOG and HOGp parameters

For HOG and HOGp descriptors three parameters were considered: window size, number of bins and normalization type. Eight window sizes were used: 1x2; 2x1; 2x2; 3x3; 4x4; 3x5; 5x3 and 5x5. Four orientation bins were used: 4 bins; 8 bins; 16 bins and 32 bins.

Let  $v$  be the unnormalized descriptor vector,  $\|v\|_k$  be its  $k$ -norm for  $k=1,2$ , and  $\epsilon$  be a small constant. The schemes are: (a) *L2-norm*,  $v \rightarrow v/\sqrt{\|v\|_2^2 + \epsilon^2}$  (b) *L1-sqrt*,  $v \rightarrow \sqrt{v/(\|v\|_1 + \epsilon^2)}$ . Four normalizations were used:

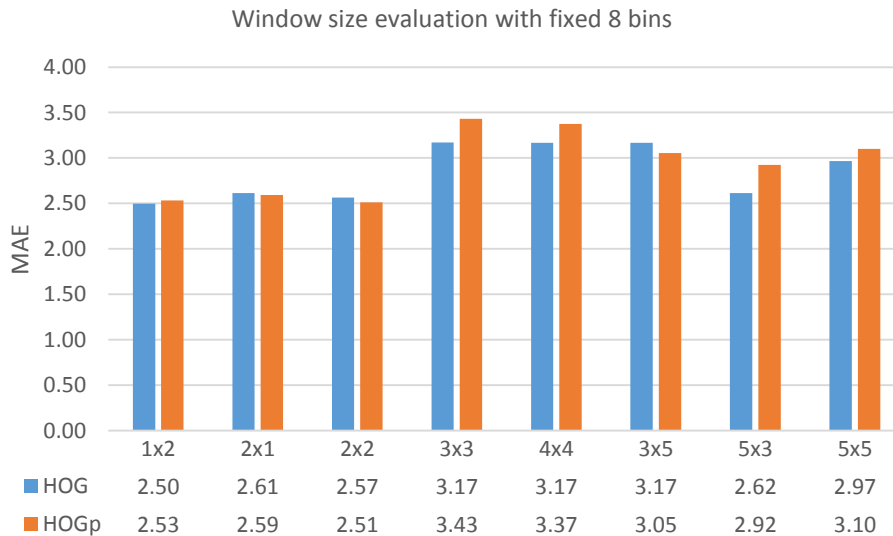
- local (by block), using *L2-norm*;
- global (for all blocks), using {Min-Max}  $\rightarrow$  {0-1};
- global, using *L1-sqrt*;
- global, using *L2-norm*.

Across all the experiments accessed in this section, the MAE results achieved when using normalization were no less than 3 times higher than without normalization. As for that, the results obtained when using normalization are not shown nor discussed in this section.

The best result with HOG descriptor was obtained when using a 1x2 window with 8 bins, with a MAE of 2.50. Nevertheless, HOGp descriptor got the best result when using a 2x1 window with 16 bins, with a MAE of 2.44. Although all 32 possible combinations of window sizes and number of bins were tested, here are only presented the combinations where these two descriptors achieved highest accuracies.

#### 4.2.1.1 Window Size

The objective of this experiment was to infer the influence of window size parameter on descriptor's accuracy. This was accomplished by fixing the number of orientation bins at 8 while changing window size from 1x2 through 5x5 and calculating the MAE for each combination (Figure 4.1).

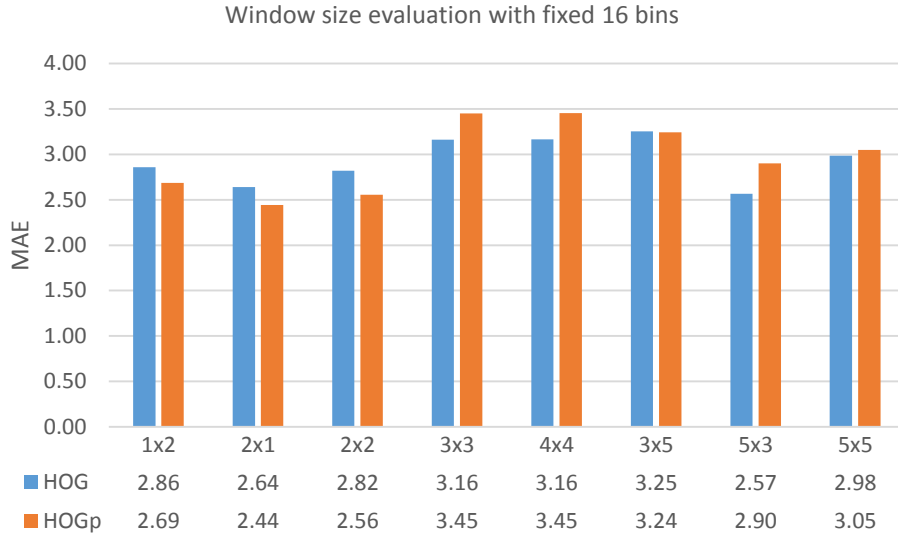


**Figure 4.1** – Experimental results of window size influence on HOG and HOGp descriptors with fixed 8 bins. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing.

It is observable that both descriptors achieve MAE values under 2.62 for smaller window sizes – 1x2, 2x1 and 2x2. Furthermore, HOG descriptor has the highest accuracy of this experiment, with 2.50 MAE with 1x2 window size. Comparing HOGp with HOG it is clear that they produce accuracies with a difference factor of

0.2–0.7 MAE, for smaller window sizes – 1x2, 2x1 and 2x2. However, for larger window sizes – 3x3, 4x4, 3x5, 5x3 and 5x5 – this factor goes up to 0.12–0.30 MAE.

When the number of bins is changed from 8 to 16 (Figure 4.2), a local minimum is introduced on 2x1 window size for both descriptors, that wasn't observable before.



**Figure 4.2** – Experimental results of window size influence on HOG and HOGp descriptors with fixed 16 bins. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing.

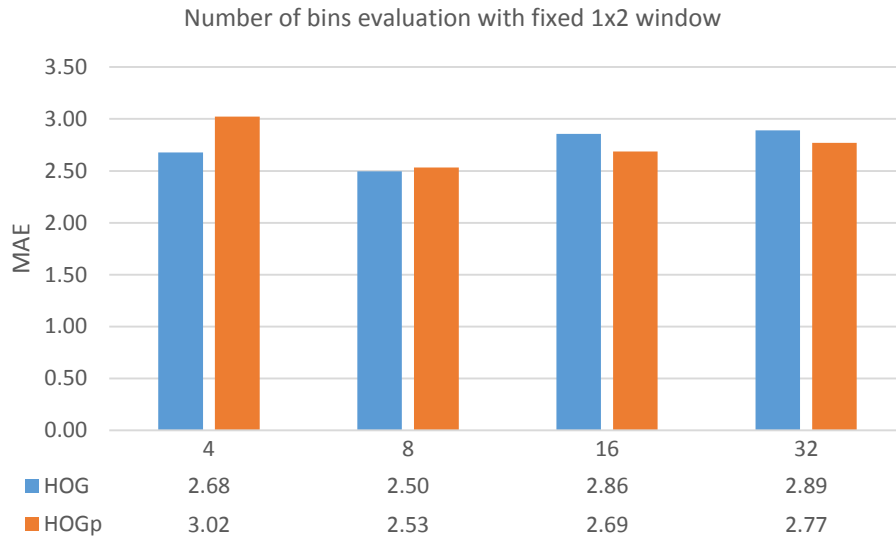
In this scenario it is clear that, for smaller window sizes – 1x2, 2x1 and 2x2 – HOGp achieved better accuracies than HOG, with a difference factor of 0.17–0.26 MAE. However, for larger window sizes – 3x3, 4x4, 3x5, 5x3 and 5x5 – HOG descriptor outstands HOGp with better average accuracy. This experiment produced the best accuracy of all the parameters tested – 2.44 MAE for HOGp vs 2.50 for HOG.

Comparing Figure 4.1 with Figure 4.2, it is observable that the variation of MAE according to window size has a similar pattern for different numbers of bins, where lower MAE values are achieved for windows 1x2, 2x1, 2x2, 5x3 and higher MAE values for 3x3, 4x4, 3x5 and 5x5.

#### 4.2.1.2 Number of bins

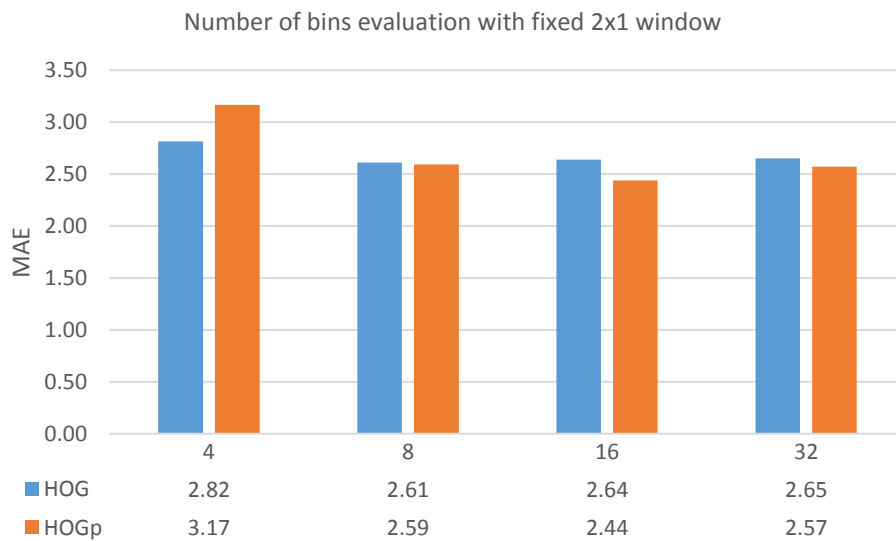
The next HOG/HOGp parameter evaluated was the number of bins. The presented experiment results were conducted with fixed window sizes of 1x2 and 2x1, combined with 4, 8, 16 and 32 bins.

Figure 4.3 shows accuracy values for both descriptors with a fixed window size of 1x2. In this experiment, HOG achieved the lowest MAE value for 8 bins – 2.50 MAE – followed by HOGp also with 8 bins – 2.53 MAE.



**Figure 4.3** – Experimental results of number of bins influence on HOG and HOGp descriptors with fixed 1x2 window size. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing.

Figure 4.4 shows accuracy values for both descriptors with a fixed window size of 2x1. In this experiment, HOGp achieved the lowest MAE value for 16 bins – 2.44 MAE – which was also the highest accuracy achieved for all the parameters' combinations. The highest MAE value were obtained for HOGp with 4 bins – 3.17 MAE – preceded by HOG with 4 bins – 2.82 MAE.



**Figure 4.4** – Experimental results of number of bins influence on HOG and HOGp descriptors with fixed 2x1 window size. MAE values are obtained by Linear Regression. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing.

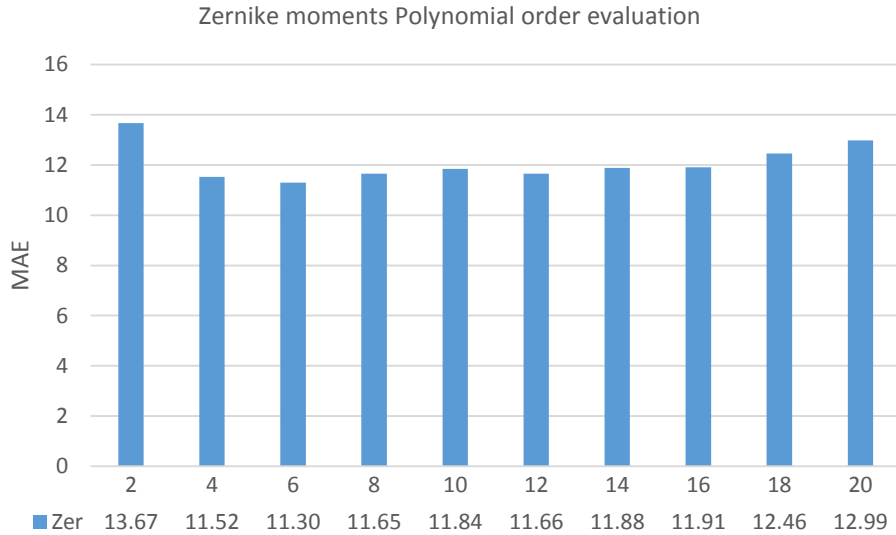
Analyzing these results it can be concluded that HOGp doesn't perform well when using 4 bins. However, when using 8, 16 and 32 bins HOGp won the accuracy test against HOG five out of six times, losing only for 1x2 window with 8 bins, by a margin of 0.03 MAE.

### 4.2.2 Zernike Moments parameters

For Zernike Moments descriptor one experiment was performed ranging the polynomial order parameter from 4 to 20 in steps of 2. It was decided to limit this parameter at 20 because, considering  $n$  the polynomial order, the number of features calculated by the descriptor ( $Zer_{feats}$ ) grows according to the quadratic function:

$$Zer_{feats} = 0.5n^2 + 1.5n + 1 \quad (4.1)$$

and the higher number of features, the higher computation time is required to calculate a count estimate.



**Figure 4.5** – Experimental results of Polynomial order variation on Zernike moments descriptor reported by Linear Regression model. Considering frames 631-1400 of UCSD dataset, the first 66% were set aside for training and the remaining 33% were used for testing.

Analyzing Figure 4.5 it is clear that the best accuracy achieved within this parameter range was for order 6 – 11.30 MAE. It can also be concluded that this descriptor alone produces results far worse than those obtained with HOG or HOGp, as all the experiments resulted in MAE values over 10.0 while for HOG and HOGp none of the experiments went across 4.0 MAE. In the next section this descriptor was combined with others in order to retrieve information whether if it was viable or not.

### 4.3 Evaluation of different combinations of descriptors

In this section are presented accuracy results for six combinations of image descriptors:

- HOGp (2x1 window size, 16 bins);
- HOG (2x1 window size, 16 bins);
- Ryan combination of descriptors [45];
- HOGp, Area (w/ perspective map), Perimeter (w/ perspective map);
- HOGp, Zer (order 6);
- HOGp, Zer, Area (w/ perspective map), Perimeter (w/ perspective map).

The parameters used for HOGp were the optimal ones on the training set, determined by the experiments discussed in Section 4.2 Although HOG parameters are not the optimal, it was decided to use the same values used in HOGp to allow further comparison of one another. Both Area and Perimeter are weighted by the UCSD dataset perspective maps, explained in Section 3.3. Ryan descriptor computes a combination of features used

by Ryan et al. in [45], which also includes Area and Perimeter weighted by the UCSD perspective maps. This descriptor was tested in order to compare the results with features used by Ryan.

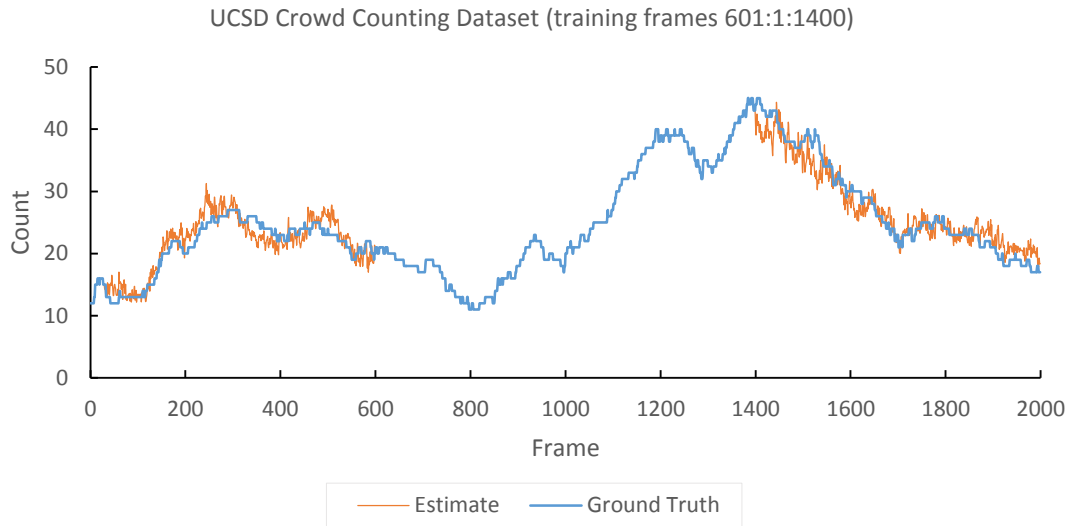
These experiments were done using full frame range of the UCSD dataset – frames 631:1400 for training, 31:600 and 1401:2000 for testing. The starting 30 frames on each set were neglected because they are used for background subtraction initialization. Table 4.2 presents MAE values for each descriptor using 9 Regression Models.

**Table 4.2** – Experimental results on UCSD dataset. Frames 631-1400 were set aside for training, and frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by 9 Regression Models. The lowest MAE value for each Regression Model is bolded and the lowest value of all is underlined.

Regression Model	Image descriptors					
	HOGp	HOG	Ryan	HOGp, A, P	HOGp, Zer	HOGp, Zer, A, P
LinearRegression	1.85	2.08	1.98	1.85	<b><u>1.74</u></b>	1.86
REPTree	2.37	2.71	2.10	<b>2.04</b>	2.44	2.12
M5P	2.01	2.16	1.89	<b>1.77</b>	1.87	1.97
Additive LinearRegression	1.85	2.08	1.98	1.85	<b><u>1.74</u></b>	1.85
Additive REPTree	2.46	2.77	2.10	<b>2.09</b>	2.43	2.15
Additive M5P	2.06	2.20	1.81	<b>1.77</b>	1.96	1.96
Additive DecisionStump	2.17	2.29	2.25	2.13	2.22	<b>2.06</b>
Bagging LinearRegression	1.85	2.08	1.98	1.85	<b><u>1.74</u></b>	1.86
Bagging REPTree	1.92	2.24	1.95	1.81	1.85	<b>1.81</b>

Analyzing these results, it stands clear that HOGp has better accuracy than HOG, as it achieves lower MAE values over each one of the Regression Models. The combination of HOGp and Zernike descriptors produced the lowest MAE value of all – 1.74 – when using Linear Regression. Furthermore, in any of the chosen Regression Models, Ryan features did not perform better than the combination of HOGp, Area and Perimeter.

Figure 4.6 presents people counting estimates for UCSD dataset, when using the optimized combination of HOGp and Zernike descriptors along with Linear Regression.



**Figure 4.6** – People counting results on UCSD dataset using HOGp and Zernike as image descriptor and Linear Regression. Frames 631-1400 were set aside for training, and frames 31-600 and 1401-2000 were used for testing.

#### 4.4 Impact of training set size

In this section are presented accuracy results for 3 different training sets, in order to evaluate the impact of the number of training frames. The test set remains at frames 1:600 and 1401:2000. The training sets used for these experiments were:

- Full (631:1:1400), for a total of 770 frames;
- 635:5:1400, for a total of 154 frames;
- 640:80:1360, for a total of 10 frames.

All the chosen 9 Regression Models were tested and these results can be viewed in Appendix A1-3 for each training set. Table 4.3 shows MAE for each training subset, considering the Regression Model that achieved the best result for each descriptor. Again, the best accuracy was obtained with HOGp combined with Zernike moments – 1.68 MAE. This is lower than the results obtained with the same descriptor with full training subset.

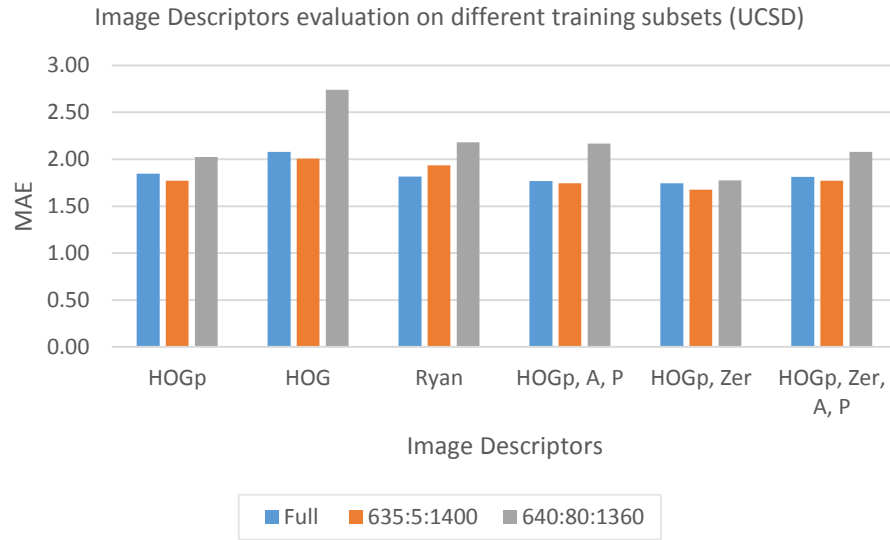
**Table 4.3** – Experimental results on UCSD dataset with 3 different training subsets: Full, 635:5:1400 and 640:80:1360. Frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. The lowest MAE value for each Regression Model is bolded and the lowest value of all is underlined.

Training subset	Image descriptors					
	HOGp	HOG	Ryan	HOGp, A, P	HOGp, Zer	HOGp, Zer, A, P
Full	1.85	2.08	1.82	1.77	<b>1.74</b>	1.81
635:5:1400	1.77	2.01	1.93	1.74	<b><u>1.68</u></b>	1.77
640:80:1360	2.02	2.74	2.18	2.17	<b>1.77</b>	2.08

In Figure 4.7 is presented the same information of Table 4.3 optimized for descriptor visual comparison. It stands clear that HOG obtained higher MAE values across all training subsets. Furthermore, the subset 635:5:1400 (154 frames) achieved optimal results for all descriptors' combinations except for Ryan.



Nonetheless, the subset 640:80:1360 (10 frames) produced the worst accuracy results for all descriptors' combinations.



**Figure 4.7** – Experimental results on UCSD dataset with 3 different training subsets: Full, 635:5:1400 and 640:80:1360. Frames 31-600 and 1401-2000 were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models.

## 4.5 Evaluation on the Future Cities dataset

This section presents experimental results for the Future Cities dataset. The counting algorithms were tested using 3 different training sets: first 25 frames, first 50 frames and 100 frames. FC dataset has 408 fully annotated frames and for each training set, the remaining frames were used for testing. All the chosen 9 Regression Models were tested and these results can be viewed in Appendix A4-6, for each training set.

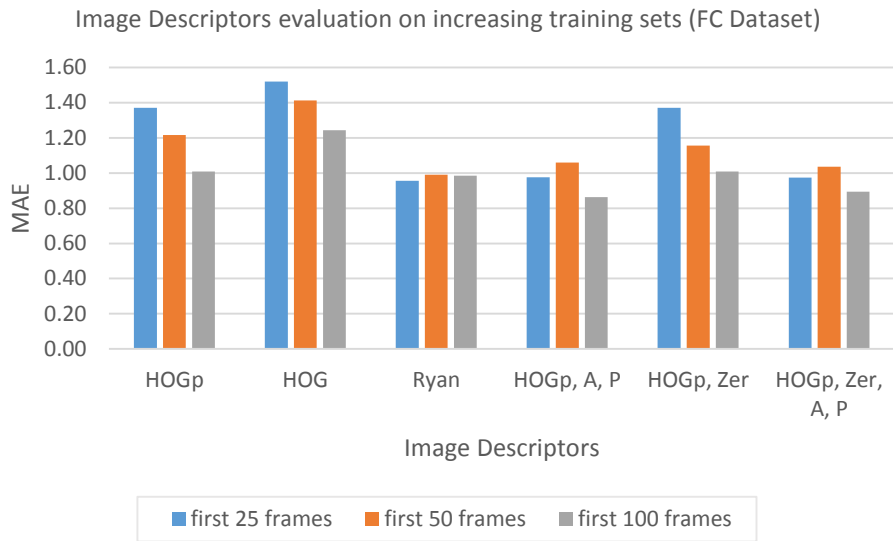
Table 4.4 shows MAE for each training subset, considering the Regression Model that achieved the best result for each descriptor. In this dataset, Ryan's descriptors obtained lower MAE values for the first two subsets, while the combination HOGp, Area and Perimeter achieved the best overall result – 0.86 MAE.

**Table 4.4** – Experimental results on FC dataset with 3 different training subsets: first 25 frames, 50 frames and the first 100 frames. The remaining dataset frames were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models. The lowest MAE value for each Regression Model is bolded and the lowest value of all is underlined.

Training set	Image descriptors					
	HOGp	HOG	Ryan	HOGp, A, P	HOGp, Zer	HOGp, Zer, A, P
first 25 frames	1.37	1.52	<b>0.96</b>	0.98	1.37	0.97
first 50 frames	1.22	1.41	<b>0.99</b>	1.06	1.16	1.04
first 100 frames	1.01	1.24	0.98	<u><b>0.86</b></u>	1.01	0.89

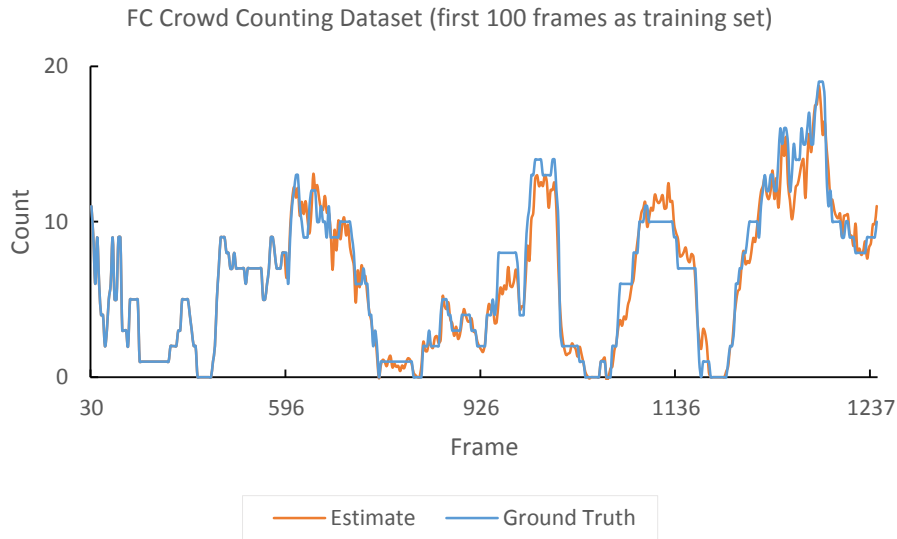
Using this dataset, Zernike descriptor does not improve HOGp as it did with UCSD dataset, in fact it only decreases MAE by a maximum margin of 0.02, when both are combined with Area and Perimeter. However, Area and Perimeter now have a much higher impact when combined with HOGp and Zernike, decreasing MAE by a minimum margin of 0.12.

In Figure 4.8 is presented the same information of Table 4.4 optimized for descriptor visual comparison. It stands clear that HOG obtained higher MAE values across all training subsets, while descriptor combinations that include Area and Perimeter produce lower MAE values.



**Figure 4.8** – Experimental results on FC dataset with 3 different training subsets: first 25 frames, 50 frames and the first 100 frames. The remaining dataset frames were used for testing. 6 image descriptor combinations were considered and MAE values are reported by the lowest MAE value achieved by the chosen 9 Regression Models.

Figure 4.9 presents people counting estimates for FC dataset, when using the optimized combination of HOGp, Area and Perimeter along with Linear Regression with Bagging.



**Figure 4.9** – People counting results on FC Dataset using HOGp, Area and Perimeter as image descriptors and Linear Regression with Bagging. The first 100 frames were set aside for training and the remaining were used for testing.

## 4.6 Discussion

This chapter reports the most relevant results achieved throughout thesis' development. Section 4.1 shows the starting point to why Zernike moments and Histogram of Oriented Gradients were selected as possible image descriptors for the people counting algorithm. However, Relief-F is a feature selection algorithm that analyzes features without considering relation between them, so it doesn't reckon the image descriptor as whole. For this same reason, Zernike moments was ranked first with Relief-F, but this descriptor alone retrieved MAE values far worse than HOG or HOGp, as depicted in Section 4.2.2. An alternative to the chosen ranking method, would be to use Correlation Feature Selection (CFS) or Wrapper. Another possible experiment would be to divide feature vectors into smaller ones, by taking out the features that were worse ranked by Relief-F, in order to see if the MAE converges for a specific subset of features. Relief-F was ultimately chosen due to the large number of attributes considered, because if it was used a selection algorithm based on groups of attributes, the computational effort would be impractical under the thesis circumstances.

Section 4.2, presents a study of HOG, HOGp and Zernike moments parameters. Using normalization in HOG and HOGp, produced results with high MAE values, when compared to unnormalized HOG and HOGp. This was somewhat expected because by normalizing gradient vectors they become invariant to multiplications of the pixel values. This is beneficial for the original focus of HOG, people detection, in order to make it invariant to illumination changes. However, for people counting purposes, normalization has a negative effect, because it scales gradients' magnitude for the ROI, losing information that is important to infer the number of people.

By evaluating HOG and HOGp descriptors with different parameters, it was determined that smaller window sizes describe people density better than larger window sizes. This is beneficial because smaller windows result in fewer attributes than larger ones, which reduces the processing time of the regression algorithms. On the other hand, studying different number of orientation bins did not produce such conclusive results. Calculating HOG and HOGp with only 4 bins produced bad results in both situations presented. Using 8, 16 and 32 bins results were relatively close, so it was chosen to fix number of bins at 8 for further experiments, once again because it generates fewer features.

From Section 4.3 to 4.5, the proposed descriptor HOGp reports better results than HOG in every single experiment, winning by an average MAE margin of 0.2, which is a quite significant error decay. For the UCSD dataset, with a training set of 10 frames, this margin went as far as 0.72 (recall Section 4.4).

Table 4.5 compares the best obtained results with State of the Art systems, for the UCSD dataset. The proposed approach outperforms Kong [12] and Chan [14] systems and competes with Lempitsky [35] and Ryan [45]. Although Ryan reports better results for some training sets, bear in mind that they optimized all the components of the algorithm, including background subtraction and inference processes, while this work was focused on image descriptors and feature extraction. Replicating Ryan's descriptors to use with the background subtraction and regression models from this thesis, the proposed image descriptors outperformed Ryan's in almost every scenario.

**Table 4.5** – Testing results on the UCSD dataset. Frames 1-600 and 1401-200 were used for testing. Results in bold correspond to the proposed approach.

<b>Systems</b>	<b>Training subset</b>	<b>MAE</b>
Kong, linear	All	1.92
Kong, neural network (5 runs)	All	2.47
Chan, away+towards	All	1.95
Chan, all	All	1.95
<b>Proposed</b>	<b>631:1:1400</b>	<b>1.74</b>
Lempitsky	605:5:1400	2.02
<b>Proposed</b>	<b>635:5:1400</b>	<b><u>1.68</u></b>
Lempitsky	640:80:1360	1.70
Ryan, no tracking	610:80:1330	1.79
Ryan, no tracking	640:80:1360	1.33
<b>Proposed</b>	<b>640:80:1360</b>	<b>1.77</b>

Section 4.5, presents the results for the FC dataset. For this dataset, descriptors that use area and perimeter achieve better MAE values than those who do not. This is mainly due to the lower number of people per frame and their sparse distribution, when compared to UCSD dataset where crowd density is higher and occlusion is frequent. When individuals are fully distinguishable (FC dataset), Area and Perimeter gain importance as features for people counting. On the other hand, if occlusion and crowded areas are more frequent (UCSD dataset), HOG, HOGp and Zernike moments are better descriptors for estimating the number of people.

Ryan features outperformed the proposed descriptors for the first two training sets of 25 and 50 frames, while losing for the third training set of 100 frames. However, one frame from this dataset contributes with much fewer training instances than one frame from UCSD dataset, because the crowd density is lower so the number of blobs per frame is also low. For instance, 100 FC frames contribute with 480 training instances, while the same number of 480 training instances is achieved with only 20 frames from UCSD dataset. In conclusion, the proposed algorithm outperforms Ryan features for adequate training sets, because each descriptor produces feature vectors with more attributes than Ryan’s descriptor, so more training instances are needed in order to surpass Ryan’s descriptor.

## Chapter 5

# Conclusions and Future work

In this thesis, a people counting algorithm was successfully implemented, trained and tested and a new image descriptor was proposed. Although camera calibration was not used, the algorithm still managed to perform well, even under different capture scenes. Several experiments were conducted using this algorithm, in order to evaluate its performance with different image descriptors, datasets and regression models.

Recalling the objectives defined in Section 1.3, the first stage was to design a vision based method for counting people. Literature reviewed in Chapter 2, provided deeper knowledge on the theme, that lead to a smoother transition to design stage, and better integration with the problem in hands. Adapting methods used in some remarkable past works, it was possible to design an algorithm capable of learning and estimating the number of pedestrian on a given capture environment.

The proposed counting algorithm uses image descriptors to extract local features from foreground segments. Different regression models were trained with the extracted feature vectors and their corresponding ground truth count, in order to infer the number of people in newer test instances. Several image descriptors were used in this work, including one distinguishable descriptor proposed by Dalal and Triggs, named Histograms of Oriented Gradients. In this thesis is proposed an extension to HOG, using density maps to normalize the descriptor for perspective effects. This is especially useful for outdoor scenarios, because the camera cannot be placed directly above people's head and distant pedestrians are segmented into fewer image pixels than closer ones. The results presented in Chapter 4, show that this extension, denoted HOGp, was indeed a contribution, because the counting error was reduced in several performed experiments.

Both HOG and HOGp were optimized to achieve the best results. Combining these descriptors with Zernike moments, blob area and blob perimeter, the final counting errors were further reduced. The study of these descriptors under several experiment conditions and different regression models constitutes another contribution of the thesis. In fact, the introduction of Zernike moments descriptor as a local feature extractor is by itself a contribution, as it has not been used for people counting in previous works.

The study of image descriptors for people counting is still and incomplete task. The former work can be extended and the algorithm can also be further enhanced. For instance, instead of using Relief-F to for feature selection, an algorithm based of grouped attributes could be used, in order to judge descriptors as a whole and not by individual feature power. This could lead to new studies, on different image descriptors than the ones considered in this thesis. Furthermore, if the computational resources were good enough, a tracking module could be implemented and added to the algorithm. This could turn the algorithm more robust and even extend its capabilities, by allowing not only accurate crowd size estimations, but also crowd dynamics statistics.

The solution presented in this thesis uses a single camera to capture video with low resolution and low frame rate. In addition, this approach does not need images with a specific colour space, as it was designed to use grayscale images as input. The chosen regression models are also simple, and easy to implement in other computer languages. For these reasons, the proposed solution can be implemented in systems with low computational resources.

Another possible way to evolve the thesis would be to implement the code on a device already in use in the Future Cities Project, the Raspberry-Pi. In fact, the FC dataset was captured and constructed with a Raspberry-Pi, running the background subtraction algorithm described in Section 3.2. This unit is a low-cost, mini single-board computer that can perform general tasks as a usual PC with lower processing speed and memory capacity, which opens the possibility to estimate the number of people on-site. Because the video would be processed locally, without any recording, streaming or even a dedicated server, privacy concerns would be minimized and this integrated system could be deployed on strategic urban locations, covering a large area of a city.

The possibilities of using crowd density data are inspiring. For instance, this information can help city planners to identify locations in need of public transportation or can provide safety and surveillance by detecting abnormal behaviors. In conclusion, it can be used, solely or combined with other statistical data, to push the inevitable city growth in the right sustainable direction.

## Appendix A – Error tables

### A1 MAE for UCSD (631:1:1400 as training set)

HOGp	
LinearRegression	1.847552991
REPTree	2.34627094
M5P	2.007508547
Additive LinearRegression	1.847552991
Additive REPTree	2.462930769
Additive M5P	2.062833333
Additive DecisionStump	2.175417949
Bagging LinearRegression	1.847324786
Bagging REPTree	1.920853846

HOG	
LinearRegression	2.077387179
REPTree	2.707197436
M5P	2.157246154
Additive LinearRegression	2.077387179
Additive REPTree	2.768040171
Additive M5P	2.200570085
Additive DecisionStump	2.294007692
Bagging LinearRegression	2.084540171
Bagging REPTree	2.244011966

Ryan	
LinearRegression	1.978618803
REPTree	2.100584615
M5P	1.886447863
Additive LinearRegression	1.978618803
Additive REPTree	2.099810256
Additive M5P	1.815200855
Additive DecisionStump	2.250996581
Bagging LinearRegression	1.981315385
Bagging REPTree	1.950021368

HOGp + A + P	
LinearRegression	1.848082906
REPTree	2.042478632
M5P	1.767767521
Additive LinearRegression	1.848082906
Additive REPTree	2.089515385
Additive M5P	1.770547863
Additive DecisionStump	2.130542735
Bagging LinearRegression	1.848418803
Bagging REPTree	1.815082906

HOGp + Z	
LinearRegression	1.743624786
REPTree	2.43735641
M5P	1.870609402
Additive LinearRegression	1.743624786
Additive REPTree	2.42842735
Additive M5P	1.960453846
Additive DecisionStump	2.220417094
Bagging LinearRegression	1.745567521
Bagging REPTree	1.85148547

HOGp + Z + A + P	
LinearRegression	1.864254701
REPTree	2.123483761
M5P	1.971094017
Additive LinearRegression	1.848082906
Additive REPTree	2.147871795
Additive M5P	1.961240171
Additive DecisionStump	2.059666667
Bagging LinearRegression	1.859035043
Bagging REPTree	1.813664103

## A2 MAE for UCSD (635:5:1400 as training set)

HOGp	
LinearRegression	1.790025641
REPTree	2.397535897
M5P	1.942987179
Additive LinearRegression	1.790025641
Additive REPTree	2.713006838
Additive M5P	1.94781453
Additive DecisionStump	2.695458974
Bagging LinearRegression	1.771168376
Bagging REPTree	1.922037607

HOG	
LinearRegression	2.023273504
REPTree	2.855418803
M5P	2.302826496
Additive LinearRegression	2.023273504
Additive REPTree	2.896762393
Additive M5P	2.257013675
Additive DecisionStump	2.663902564
Bagging LinearRegression	2.007860684
Bagging REPTree	2.3763

Ryan	
LinearRegression	1.995119658
REPTree	2.470163248
M5P	1.930636752
Additive LinearRegression	1.995119658
Additive REPTree	2.32757265
Additive M5P	1.934045299
Additive DecisionStump	2.47692906
Bagging LinearRegression	1.97494188
Bagging REPTree	2.016192308

HOGp + A + P	
LinearRegression	1.832994872
REPTree	2.444381197
M5P	1.74354359
Additive LinearRegression	1.832994872
Additive REPTree	2.219405128
Additive M5P	1.766735897
Additive DecisionStump	2.498403419
Bagging LinearRegression	1.825747863
Bagging REPTree	2.037163248

HOGp + Z	
LinearRegression	1.689715385
REPTree	2.374674359
M5P	1.846478632
Additive LinearRegression	1.689715385
Additive REPTree	2.507968376
Additive M5P	1.806087179
Additive DecisionStump	2.57555812
Bagging LinearRegression	1.676207692
Bagging REPTree	1.85927094

HOGp + Z + A + P	
LinearRegression	1.865909402
REPTree	2.426689744
M5P	1.865909402
Additive LinearRegression	1.865909402
Additive REPTree	2.255723077
Additive M5P	1.772784615
Additive DecisionStump	2.445005983
Bagging LinearRegression	1.85022906
Bagging REPTree	1.964617094



### A3 MAE for UCSD (640:80:1360 as training set)

HOGp	
LinearRegression	2.455306838
REPTree	6.095699145
M5P	2.023722222
Additive LinearRegression	2.455306838
Additive REPTree	6.095699145
Additive M5P	2.167183761
Additive DecisionStump	4.308476068
Bagging LinearRegression	2.483764957
Bagging REPTree	2.560554701

HOG	
LinearRegression	2.80494359
REPTree	6.095699145
M5P	2.741394017
Additive LinearRegression	2.80494359
Additive REPTree	6.095699145
Additive M5P	2.741394017
Additive DecisionStump	4.781047863
Bagging LinearRegression	2.893994872
Bagging REPTree	3.445828205

Ryan	
LinearRegression	2.182117094
REPTree	6.095699145
M5P	2.182117094
Additive LinearRegression	2.182117094
Additive REPTree	6.095699145
Additive M5P	2.182117094
Additive DecisionStump	4.050364957
Bagging LinearRegression	2.224895726
Bagging REPTree	3.052439316

HOGp + A + P	
LinearRegression	2.340208547
REPTree	6.095699145
M5P	2.16765812
Additive LinearRegression	2.340208547
Additive REPTree	6.095699145
Additive M5P	2.310951282
Additive DecisionStump	4.510918803
Bagging LinearRegression	2.550525641
Bagging REPTree	2.694694872

HOGp + Z	
LinearRegression	2.451562393
REPTree	6.095699145
M5P	1.773703419
Additive LinearRegression	2.451562393
Additive REPTree	6.095699145
Additive M5P	2.068573504
Additive DecisionStump	4.298644444
Bagging LinearRegression	2.642125641
Bagging REPTree	2.576911966

HOGp + Z + A + P	
LinearRegression	2.317713675
REPTree	6.095699145
M5P	2.079958974
Additive LinearRegression	2.317713675
Additive REPTree	6.095699145
Additive M5P	2.207398291
Additive DecisionStump	4.754211966
Bagging LinearRegression	2.336051282
Bagging REPTree	2.739491453

#### A4 MAE for FC (first 25 frames as training set)

HOGp	
LinearRegression	1.589201044
REPTree	1.797958225
M5P	1.437326371
Additive LinearRegression	1.589201044
Additive REPTree	1.797958225
Additive M5P	1.369809399
Additive DecisionStump	1.46313577
Bagging LinearRegression	1.453691906
Bagging REPTree	1.714624021

HOG	
LinearRegression	1.696023499
REPTree	1.837642298
M5P	1.523143603
Additive LinearRegression	1.696023499
Additive REPTree	1.813597911
Additive M5P	1.659660574
Additive DecisionStump	1.519425587
Bagging LinearRegression	1.673143603
Bagging REPTree	1.78113577

Ryan	
LinearRegression	0.955986945
REPTree	1.618093995
M5P	0.955986945
Additive LinearRegression	0.955986945
Additive REPTree	1.618093995
Additive M5P	0.955986945
Additive DecisionStump	1.218010444
Bagging LinearRegression	0.96083812
Bagging REPTree	1.571373368

HOGp + A + P	
LinearRegression	1.322798956
REPTree	1.692120104
M5P	0.976409922
Additive LinearRegression	1.322798956
Additive REPTree	1.646788512
Additive M5P	1.06348564
Additive DecisionStump	1.641712794
Bagging LinearRegression	1.345610966
Bagging REPTree	1.674174935

HOGp + Z	
LinearRegression	1.634887728
REPTree	1.796553525
M5P	1.370652742
Additive LinearRegression	1.634887728
Additive REPTree	1.807934726
Additive M5P	1.472399478
Additive DecisionStump	1.611430809
Bagging LinearRegression	1.641263708
Bagging REPTree	1.733835509

HOGp + Z + A + P	
LinearRegression	1.306569191
REPTree	1.698300261
M5P	0.973328982
Additive LinearRegression	1.306569191
Additive REPTree	1.624673629
Additive M5P	1.118190601
Additive DecisionStump	1.565443864
Bagging LinearRegression	1.583644909
Bagging REPTree	1.692613577

## A5 MAE for FC (first 50 frames as training set)

HOGp	
LinearRegression	1.312254902
REPTree	2.153403361
M5P	1.215820728
Additive LinearRegression	1.312254902
Additive REPTree	2.153403361
Additive M5P	1.296131653
Additive DecisionStump	1.792910364
Bagging LinearRegression	1.268422969
Bagging REPTree	1.909616246

HOG	
LinearRegression	1.466918768
REPTree	2.062078431
M5P	1.495344538
Additive LinearRegression	1.466918768
Additive REPTree	2.062689076
Additive M5P	1.411994398
Additive DecisionStump	1.588551821
Bagging LinearRegression	1.488044818
Bagging REPTree	1.914868347

Ryan	
LinearRegression	0.990294118
REPTree	1.960733894
M5P	0.990294118
Additive LinearRegression	0.990294118
Additive REPTree	1.560915966
Additive M5P	0.990294118
Additive DecisionStump	1.425288515
Bagging LinearRegression	1.066557423
Bagging REPTree	1.730263305

HOGp + A + P	
LinearRegression	1.066313725
REPTree	2.03167507
M5P	1.101386555
Additive LinearRegression	1.066313725
Additive REPTree	2.153403361
Additive M5P	1.059868347
Additive DecisionStump	1.423882353
Bagging LinearRegression	1.119417367
Bagging REPTree	1.905792717

HOGp + Z	
LinearRegression	1.187941176
REPTree	2.142661064
M5P	1.156193277
Additive LinearRegression	1.187941176
Additive REPTree	2.142661064
Additive M5P	1.159812325
Additive DecisionStump	1.523252101
Bagging LinearRegression	1.1992493
Bagging REPTree	1.927507003

HOGp + Z + A + P	
LinearRegression	1.039686275
REPTree	2.025789916
M5P	1.035380952
Additive LinearRegression	1.039686275
Additive REPTree	2.142661064
Additive M5P	1.165635854
Additive DecisionStump	1.591212885
Bagging LinearRegression	1.147366947
Bagging REPTree	1.917109244

## A6 MAE for FC (first 100 frames as train)

HOGp	
LinearRegression	1.084814332
REPTree	1.344286645
M5P	1.009078176
Additive LinearRegression	1.084814332
Additive REPTree	1.322032573
Additive M5P	1.027214984
Additive DecisionStump	1.42795114
Bagging LinearRegression	1.055811075
Bagging REPTree	1.474690554

HOG	
LinearRegression	1.298188925
REPTree	1.651214984
M5P	1.244061889
Additive LinearRegression	1.298188925
Additive REPTree	1.734410423
Additive M5P	1.26081759
Additive DecisionStump	1.523394137
Bagging LinearRegression	1.305459283
Bagging REPTree	1.480996743

Ryan	
LinearRegression	1.031211726
REPTree	1.531035831
M5P	0.986628664
Additive LinearRegression	1.031211726
Additive REPTree	1.186403909
Additive M5P	0.983967427
Additive DecisionStump	1.133469055
Bagging LinearRegression	1.037107492
Bagging REPTree	1.233869707

HOGp + A + P	
LinearRegression	0.884387622
REPTree	1.542188925
M5P	0.928550489
Additive LinearRegression	0.884387622
Additive REPTree	1.509934853
Additive M5P	0.92185342
Additive DecisionStump	1.320625407
Bagging LinearRegression	0.862960912
Bagging REPTree	1.386824104

HOGp + Z	
LinearRegression	1.083254072
REPTree	1.491885993
M5P	1.039631922
Additive LinearRegression	1.083254072
Additive REPTree	1.532478827
Additive M5P	1.007967427
Additive DecisionStump	1.521749186
Bagging LinearRegression	1.046899023
Bagging REPTree	1.550964169

HOGp + Z + A + P	
LinearRegression	0.957964169
REPTree	1.537091205
M5P	1.051003257
Additive LinearRegression	0.957964169
Additive REPTree	1.603745928
Additive M5P	0.893833876
Additive DecisionStump	1.2797557
Bagging LinearRegression	0.914801303
Bagging REPTree	1.42662215

# References

- [1] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source Multi-scale Counting in Extremely Dense Crowd Images," *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2547–2554, Jun. 2013.
- [2] K. Kopaczewski, M. Szczodrak, a. Czyzewski, and H. Krawczyk, "A method for counting people attending large public events," *Multimed. Tools Appl.*, Aug. 2013.
- [3] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Scene invariant multi camera crowd counting," *Pattern Recognit. Lett.*, Oct. 2013.
- [4] "Future Cities: An Ecosystem for the Future." [Online]. Available: <http://futurecities.up.pt/site/>. [Accessed: 13-Feb-2014].
- [5] "European Commission. CORDIS. FP7. URL." [Online]. Available: [http://cordis.europa.eu/fp7/home\\_en.html](http://cordis.europa.eu/fp7/home_en.html). [Accessed: 13-Feb-2014].
- [6] a. Albiol, I. Mora, and V. Naranjo, "Real-time high density people counter using morphological tools," *IEEE Trans. Intell. Transp. Syst.*, vol. 2, no. 4, pp. 204–218, 2001.
- [7] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "Method of counting the passing people by using the stereo images," in *IEEE International Conference on Image Processing*, 1999, vol. 2, pp. 338–342.
- [8] D. Beymer and K. Konolige, "Real-time tracking of multiple people using stereo," *Proc. IEEE Fram. Rate Work.*, 1999.
- [9] B. Kim, G. Lee, J. Yoon, J. Kim, and W. Kim, "A Method of Counting Pedestrians in Crowded Scenes," *Proceeding ICIC '08 Proc. 4th Int. Conf. Intell. Comput. Adv. Intell. Comput. Theor. Appl. - with Asp. Artif. Intell.*, pp. 1117–1126, 2008.
- [10] V. B. Subburaman, A. Descamps, and C. Carincotte, "Counting People in the Crowd Using a Generic Head Detector," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, 2012, pp. 470–475.
- [11] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Comput. Vis. Image Underst.*, vol. 110, no. 1, pp. 43–59, Apr. 2008.
- [12] D. Kong and D. Gray, "A Viewpoint Invariant Approach for Crowd Counting," *18th Int. Conf. Pattern Recognit.*, vol. 1, pp. 1187–1190, 2006.
- [13] K. Aziz, D. Merad, B. Fertil, and N. Thome, "Pedestrian Head Detection and Tracking Using Skeleton Graph for People Counting in Crowded Environments.," *MVA*, pp. 1–4, 2011.
- [14] A. B. Chan and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," *2008 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–7, Jun. 2008.

- [15] G. Sexton, "Advances in automated pedestrian counting," in *European Convention on Security and Detection*, 1995, vol. 1995, pp. 106–110.
- [16] J. Segen, "A camera-based system for tracking people in real time," in *Proceedings of 13th International Conference on Pattern Recognition*, 1996, vol. 3, pp. 63–67 vol.3.
- [17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings. 1999 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (Cat. No PR00149)*, pp. 246–252.
- [18] S. Atev, O. Masoud, and N. Papanikolopoulos, "Practical mixtures of Gaussians with brightness monitoring," in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, pp. 423–428.
- [19] O. Masoud and N. P. Papanikolopoulos, "A novel method for tracking and counting pedestrians in real-time using a single camera," *IEEE Trans. Veh. Technol.*, vol. 50, no. 5, pp. 1267–1278, 2001.
- [20] C.-H. (Thou-H. Chen, Y.-C. Chang, T.-Y. Chen, and D.-J. Wang, "People Counting System for Getting In/Out of a Bus Based on Video Processing," *2008 Eighth Int. Conf. Intell. Syst. Des. Appl.*, pp. 565–569, Nov. 2008.
- [21] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 909–26, May 2008.
- [22] S. P. Denman, "Improved detection and tracking of objects in surveillance video." Queensland University of Technology, 16-Dec-2009.
- [23] A. J. Schofield, "A RAM based neural network approach to people counting," in *Fifth International Conference on Image Processing and its Applications*, 1995, vol. 1995, pp. 652–656.
- [24] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I-431–I-438.
- [25] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda, "People count system using multi-sensing application," in *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, 1997, vol. 2, pp. 1291–1294.
- [26] A. Tesei, A. Teschioni, C. S. Regazzoni, and G. Vernazza, "'Long-Memory' matching of interacting complex objects from real image sequences," *Time Varying Image Process. Mov. Object Recognition, Firenze*, 1996.
- [27] A. Shio and J. Sklansky, "Segmentation of people in motion," in *Proceedings of the IEEE Workshop on Visual Motion*, 1991, pp. 325–332.
- [28] G. Conrad and R. Johnsonbaugh, "A real-time people counter," in *Proceedings of the 1994 ACM symposium on Applied computing - SAC '94*, 1994, pp. 20–24.
- [29] W. Ge and R. T. Collins, "Evaluation of sampling-based pedestrian detection for crowd counting," in *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009, pp. 1–7.
- [30] N. Thome, D. Merad, and S. Miguët, "Learning articulated appearance models for tracking humans: A spectral graph matching approach," *Signal Process. Image Commun.*, vol. 23, no. 10, pp. 769–787, Nov. 2008.

- [31] J. A. Hyman, "Computer Vision Based People Tracking for Motivating Behavior in Public Spaces by," 2003.
- [32] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression.," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–77, Apr. 2012.
- [33] M. Shimosaka, S. Masuda, and R. Fukui, "Counting pedestrians in crowded scenes with efficient sparse learning," *Proceedings First ...*, pp. 27–31, 2011.
- [34] J. Zhang, B. Tan, F. Sha, and L. He, "Predicting Pedestrian Counts in Crowded Scenes With Rich and High-Dimensional Features," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1037–1046, Dec. 2011.
- [35] V. Lempitsky and A. Zisserman, "Learning To Count Objects in Images," *NIPS*, pp. 1–9, 2010.
- [36] J. Garcia, A. Gardel, I. Bravo, J. L. Lazaro, M. Martinez, and D. Rodriguez, "Directional People Counter Based on Head Tracking," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3991–4000, Sep. 2013.
- [37] A. B. Chan, M. Morrow, and N. Vasconcelos, "Analysis of crowded scenes using holistic properties," *Proc. 11th IEEE Int. Work. Perform. Eval. Track. Surveill.*, 2009.
- [38] "PETS 2006." [Online]. Available: <http://www.cvg.rdg.ac.uk/PETS2006/data.html>. [Accessed: 30-Jun-2014].
- [39] "SAIVT-QUT Crowd Counting Database - Speech, Audio, Image and Video Technologies - Confluence." [Online]. Available: <https://wiki.qut.edu.au/display/saivt/SAIVT-QUT+Crowd+Counting+Database>. [Accessed: 30-Jun-2014].
- [40] "SVCL - Crowd Counting." [Online]. Available: <http://www.svcl.ucsd.edu/projects/peoplecnt/>. [Accessed: 30-Jun-2014].
- [41] "PETS 2009." [Online]. Available: <http://www.cvg.rdg.ac.uk/PETS2009/>. [Accessed: 30-Jun-2014].
- [42] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," *Proc. 17th Int. Conf. Pattern Recognition, 2004. ICPR 2004.*, no. 2, pp. 28–31 Vol.2, 2004.
- [43] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," pp. 175–181, Aug. 1997.
- [44] "Motion Analysis and Object Tracking — OpenCV 2.4.9.0 documentation." [Online]. Available: [http://docs.opencv.org/modules/video/doc/motion\\_analysis\\_and\\_object\\_tracking.html?highlight=backgroundsubtractorMOG2#backgroundsubtractormog2](http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html?highlight=backgroundsubtractorMOG2#backgroundsubtractormog2). [Accessed: 30-Jun-2014].
- [45] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "Scene invariant crowd counting and crowd occupancy analysis," *Video Anal. Bus. ...*, pp. 161–198, 2012.
- [46] D. C. Moura and M. a Guevara López, "An evaluation of image descriptors combined with clinical data for breast cancer diagnosis.," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 8, no. 4, pp. 561–74, Jul. 2013.
- [47] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. 2008.
- [48] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Inf. theory*, 1962.

- [49] S. O. Belkasim, M. Shridhar, and M. Ahmadi, "Pattern recognition with moment invariants: A comparative study and new results," *Pattern Recognit.*, vol. 24, no. 12, pp. 1117–1138, Jan. 1991.
- [50] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Syst. Man. Cybern.*, vol. 3, no. 6, pp. 610–621, Nov. 1973.
- [51] M. M. Galloway, "Texture analysis using gray level run lengths," *Comput. Graph. Image Process.*, vol. 4, no. 2, pp. 172–179, Jun. 1975.
- [52] a. N. Marana, S. a. Velastin, L. F. Costa, and R. a. Lotufo, "Automatic estimation of crowd density using texture," *Saf. Sci.*, vol. 28, no. 3, pp. 165–175, Apr. 1998.
- [53] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 31, no. 6, pp. 645–654, 2001.
- [54] K. J. Falconer, *Fractal geometry: mathematical foundations and applications*. 1990.
- [55] N. Dalal, B. Triggs, and D. Europe, "Histograms of Oriented Gradients for Human Detection," 2005.
- [56] A. De, N. Do, and R. O. F. The, "Human detection solution for a retail store environment," 2013.
- [57] W. R. Schwartz, A. Kembhavi, D. Harwood, L. S. Davis, A. V. W. Building, and C. Park, "Human Detection Using Partial Least Squares Analysis," no. Iccv, 2009.
- [58] C. P. T. P. Anuj Mohan, "Example-Based Object Detection in Images by Components."
- [59] R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, pp. 506–513.
- [60] S. A. Velastin and A. C. Davies, "Crowd monitoring using image processing," *Electron. Commun. Eng. J.*, vol. 7, no. 1, pp. 37–47, Feb. 1995.
- [61] J. H. Friedman, "Stochastic gradient boosting," *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
- [62] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
- [63] K. Kira and L. A. Rendell, "The feature selection problem: traditional methods and a new algorithm," pp. 129–134, Jul. 1992.