Vision-based Analysis and Simulation of Pedestrian Dynamics

Author: PANAGIOTIS SOURTZINOS

Director of Studies: Dr. Dimitrios Makris

LEGION LTD

Digital Imaging Research Centre
Faculty of Science, Engineering and Computing
Kingston University
Penrhyn Road, Kingston-upon-Thames
KT1 2EE, London, U.K.

This Thesis is being submitted in partial fulfilment of the requirements of Kingston University for the Degree of Doctor of Philosophy (Ph.D.)
May 2016

London, KT1 2EE
United Kingdom

Declaration

This report is submitted as requirement for a Ph.D. Degree in the School of Computing and Information Systems (Faculty of Science, Engineering and Computing) at Kingston University. It is substantially the result of my own work except where explicitly indicated in the text.

No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other UK or foreign examination board, university or other institute of learning.

The thesis work was conducted from March 2011 to December 2015 under the supervision of Dimitrios Makris, in the Digital Imaging Research Centre (DIRC) of Kingston University in London.

Kingston-upon-Thames, London, United Kingdom.

Copyright Statement

Acknowledgements

Since every step in life comes after all the previous ones, which lead to this present step, there are many people I would have to thank for their contributions to my trip in life. Indeed so many people and so many things, that I would have to thank life itself in order to be true. For this specific step though in my life, I would have first of all to thank my supervisor, Dr Dimitrios Makris, who without his support, insights, assistance and push I wouldn't have been able to finish my PhD degree. Dimitri thank you from the bottom of my heart for your help! The other people I would have to thank are the people of Legion. Especially Dr Vassilis Zachariadis, who was the main driving force behind the simulation chapter in this thesis, and who made possible for me to explore new, uncharted territories in the land of my research interests. Other people from Legion who I would like to thank are James, Martin, Justyna and Fred who provided to me a very pleasant and supportive environment. Finally I would like to thank all the people in Kingston University, students and staff, who provided to me a stimulating, vibrant, colorful and welcoming environment in which I was able to flower, physically, mentally and spiritually. It was a long trip, which taught me that the trip itself is the goal and not the destination.

Abstract

The aim of this thesis is to examine the applicability of computer vision to analyze pedestrian and crowd characteristics, and how pedestrian simulation for shopping environments can be driven from the visual perception of the simulated pedestrians.

More specifically, two frameworks for pedestrian speed profile estimation are designed and implemented. The first address the problem of speed estimation for people moving parallel to the image plane on a flat surface, while the other tries to estimate the speed of people walking on stairs moving while their trajectories are being perpendicular on the image plane. Both approaches aim to localise the foot of the pedestrians, and by identifying their steps measure their speed.

Except from measuring the speed of pedestrians a crowd counting system using Convolutional Neural Networks is created by exploiting the background spatial persistence of a whole image in the temporal domain, and furthermore by fusing consecutive temporal counting information the system further refines its estimates.

Finally a novel memory-free cognitive framework for pedestrian shopping behaviour is presented where the simulated pedestrians use as route choice model their visual perception. Agents moving in an environment and equipped with an activity agenda, use their vision to select not only their root choices but also the shops that they visit.

Table of Contents

List of Figures

List of Abbreviations

Acronym	Definition
2D	Two Dimensional
CNN	Convolutional Neural Networks
CVA	Change Vector Analysis
DCM	Discreet Choice Models
EM	Expectation Maximisation
GIS	Geographical Information System
GP	Gaussian Processes
HOG	Histogram of Oriented Gradients
HLS	Heal Localisation System
KLT	Kanade Lucas Tomasi
LTA	Linear Trajectory Avoidance
MAE	Mean Absolute Error
MDE	Mean Deviation Error
MESA	Maximum Excess of SubArrays
MOG	Mixture Of Gaussians
MSE	Mean Square Error
MSME	Mean Shift Mode Estimation
MSPF	Mean Shift Particle Filter
MuHAVI	MUlticamera Human Action VIdeo
RNN	Recurrent Neural Network
SIFT	Scale-Invariant Feature Transform
SLS	Step Localisation System
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
VOA	Valuation Office Agency

Publications

Sourtzinos, Panagiotis, Dimitrios Makris, and Paolo Remagnino. "Highly accurate estimation of pedestrian speed profiles from video sequences." In*Innovations in Defence Support Systems–3*, pp. 71-81. Springer Berlin Heidelberg, 2011.

Sourtzinos, Panagiotis, Dimitrios Makris, James Amos, Vassilis Zachariadis. "Modelling Pedestrian Shopping Behaviour". Symposium on Applied Urban Modelling, Cambridge, UK, 24-26 June 2015

CHAPTER ONE

1. Introduction

In ancient Greece, visitors to the temple of Apollo at Delphi, while being at the forecourt and before entering the temple to receive a prophecy from the oracle, they could see and read the inscribed Delphic maxim 'Know thyself'. Although the actual meaning of this phrase might have been misunderstood by the majority of people, still even the blunt interpretation of the phrase, points to the same direction, which is the search for the truth. This inscription was a constant reminder for people to look within themselves, or to place it in a more scientific way, to research the researcher, for the quest of the truth, however the practical application of it was to look for the truth in the outer world and to examine the various phenomena for their causes and their effects.

Knowing the truth of how an object behaves, allows us to create a model for it that replicates its behaviour. This simulated object then can assist us to examine the emerging behaviour of more complex phenomena which is caused by the interaction of this object with its environment. In the case where our phenomenon under examination is the behaviour of people, and more particularly how they move into the space, the truth that is asked to be learned includes the understanding of the kinetics and preferences of people's movement, and the interaction of them with the environment. Thus by infusing to a pedestrian model true, real life, observed measurements of various characteristics, such as how fast are pedestrians moving or how many pedestrians are observed occupying a space, assist us to generate a closer to truth model. Moreover, although the behaviour of pedestrians and the route choices to their journey are affected by various factors, some well-defined like the structure of the environment, some are hidden and independent for each person, a common factor however for a pedestrian's choices is the perception of vision with which the environment is understood.

In Section 1.1 of this chapter the aims and objectives of the thesis are presented, while in 1.2 the contributions of this research are listed. Finally in Section 1.3 the structure of the thesis is discussed providing a brief summary for each following chapter.

1.1 Aims and Objectives

This research is a product of collaboration between Kingston University and Legion Ltd [82]. Legion's expertise is on modelling pedestrian movement in the space. The models they use are inspired by the true movement of people in the space, however they lack the capability to perceive real world measurements to adjust their behaviour based on the dynamics of new information received. Moreover the route choices of the pedestrians are mostly deterministic and do not take into consideration on how the structure of the environment is perceived by the people. The aim of this thesis is to examine ways of measuring real world pedestrian characteristics by using computer vision, and to investigate the correlation between people's movement in a complex environment and their visual understanding of the space. The objectives needed to fulfil this aim are:

- Design and develop a pedestrian speed estimation framework using computer vision.
- Design and develop a pedestrian counting system using computer vision.
- Design and develop a model of pedestrian movement for pedestrian route and action choices, in a complex multiple attractor environment, based on pedestrian visual perception.

The above objectives were necessary for Legion. Estimating speed profiles for pedestrians based on their observed motion, as mentioned before, brings a simulation model closer to reality, since an agent's speed in a simulation environment will be modelled using real world data. Furthermore a counting system, can provide origin/destination data on how many people enter and exit the simulation environment and from which points they do it, as well as, with estimating the probability of pedestrians choosing specific routes. Finally by developing a model of agents moving in multiple attractor environment is a first step in creating a baseline memory-free cognitive framework for simulating pedestrian shopping behaviour. Moreover the combination of the objectives of this thesis it is an initial step for automating the calibration procedure of a simulation model using online real world measurements as driving force of the calibration process.

1.2 Contributions

The first contribution of this thesis is the creation of a framework for accurate estimation of pedestrian speed profiles. The accuracy is achieved by localizing the foot of the pedestrians and by using temporal information identifies the timestamps of each step. Although speed estimation software exists the accuracy that achieves is far from what is needed by pedestrian simulation software. We present two speed estimation solutions, one for observing pedestrians walking on flat surface parallel to the image plane, while the other observes pedestrians walking on stairs perpendicular to the image plane.

The second contribution of this work is the use of Convolutional Neural Networks for people counting from fixed cameras, using whole images for training and its extension in the temporal domain. People counting using Convolutional Neural Networks has been restricted in the development of location invariant methodologies. The use of whole image information containing the full spatial relationship of the background unchanged through the temporal domain seems to be appropriate to learn the relationship between a scene specific background and the existence of people in it. Moreover the complete understanding of the relationship in a frame leads to the examination of the transferring of knowledge between consecutive frames in the temporal domain.

The third contribution of this thesis is the development of a new algorithm for estimating the isovist [9]. Isovist is the polygon in a simulation environment that contains all the visible space from an observer standing on a point in the environment. The common way of generating isovists is by using ray casting from one point to the whole of the environment in order to understand the available space a pedestrian's eyesight has access to. This computational expensive function can be replaced by examining the relationship of the pedestrian's position in the environment with the changes in the space continuity of the environment due to the presence of the various structures in it.

Finally, the fourth contribution of this thesis is the development of a pedestrian simulation framework for pedestrian shopping behaviour, using as input for the route choice model of the pedestrians the conceptual understanding of the environment, generated by high level features viewed from pedestrians' view. Multiple attractor simulation environments, where a pedestrian has a plethora of action choices, are mainly focused in modelling the desires and needs that a pedestrian has to satisfy, and the movement of pedestrians is dictated by

models which do not take of consideration of pedestrians' real visual information. The existing models, although they do incorporate environmental topological information, they don't infuse the actual visual perception of the pedestrians, as they wander in the environment, into the route and target selection choices mechanics.

1.3 Structure of the thesis

This main part of this thesis is comprised of three different but interrelated technical chapters, each one providing a solution for a specific problem. In chapter 2 a framework for accurate pedestrian speed profile estimation is presented. Two different systems are created; each one trying to measure the speed of individuals and to generate a speed profile distribution based on the observations, but with different relationships between camera angle and people projected movement on the image plane. The systems are validated against the estimation of a tracker and the ground truth observations.

In chapter 3, a people counting system for a fixed camera, developed using Convolutional Neural Networks is presented. The system is harnessing the whole image information, retaining its global spatial structure, and is compared against two other CNN systems proposed which instead use location invariant information for their implementation. Moreover the application for such network in the temporal domain is proposed for improving the precision of counting.

In chapter 4 a framework for pedestrian simulation in multiple attractor environments is presented. The environmental as well as the pedestrian model allow the creation of itineraries of people activities, and the pedestrian movement and choice model is based on the visual perception of the pedestrians. The visual perception allows a pedestrian to understand the spatial characteristics of the various structures in the environment, as well as the relationships that these form with each other. The route choice model of the framework is then validated against the ground truth and a proposed method of the literature.

Finally in chapter 5 the conclusions and future work are presented.

CHAPTER TWO

2. Accurate Pedestrian Speed Estimation

2.1 Introduction

Pedestrian simulation is increasingly being used in the design and optimization of public spaces such as transport terminals, sport, entertainment and leisure venues, shopping centres, commercial and public buildings and venues for major international events such as the Olympics. To accurately simulate pedestrian behaviour and crowd dynamics in such environments, simulation tools must be calibrated and validated using precise real world data [12]. A key determinant of crowd behaviour is the preferred walking speed of individuals, i.e. their constant speed of unimpeded walking. However walking speed has been shown to vary by context and region [78] thus appropriate speed profiles are required for each study.

The most common approach to estimate the preferred walking speed of a pedestrian is for a human operator to examine video sequences and manually mark the pedestrians' positions in each frame of the video. Pedestrians' sequential locations are analysed and the preferred speeds are estimated. The human operator only labels pedestrians that walk unimpeded. Accurate pedestrians' positions are derived by pinpointing their feet locations, as they allow unambiguous estimate of their ground speed. Furthermore, any pedestrians with varying speed are filtered out from the data. However, manual techniques are time consuming, resource intensive and error prone. Therefore, automated video analysis that can deliver a high degree of speed accuracy (e.g. at least 90% of speed estimations with error less than 10%) is highly desirable. Moreover, automatic pedestrian speed estimation should include only measurements of pedestrians who walk unimpeded and with constant speed. Unimpeded pedestrians choose a walking speed which is constant and minimizes their energy consumption [13][92][98]. Therefore in order to estimate a speed profile distribution, the observations of multiple pedestrians walking in their preferred speed are

required. In a microsimulation environment, where each pedestrian entity is represented by an agent, the speed of an agent is randomly sampled from a speed profile. Of course, this speed is dependent on the environment. For example a stroll in a park, has different purpose for a person, than walking while commuting and thus the speed of the pedestrian may vary. Thus for each simulating environment a different pedestrian speed profile is required.

An example of applying computer vision methods to calibrate a pedestrian simulation software was presented in [136]. Specifically, an off-the-shelf pedestrian tracker was used to estimate pedestrian speeds, which was then used for calibrating the PEDFLOW [73] microscopic pedestrian simulation software.

Many methods have been proposed for automatic localization/tracking of pedestrians [40] [140], and they tend to approximate the pedestrian's location on the image with the centre of gravity, or a bounding box or an ellipse. However, a more accurate representation may be based on estimating the speed of a specific body part, taking into account the biomechanics of human walking. In this chapter, two solutions are presented for accurate speed estimation, based on the estimation of foot locations. Section 2.2 discusses the relevant literature review as well as the background information needed for this work. In Section 2.3, two systems are proposed based on the camera view point: a) the heel localization system (HLS), which is applied when the camera view is sideways to pedestrian movement and b) the step localisation system (SLS) (i.e. pedestrians are viewed from upfront on stairs). The proposed methods are evaluated using three video datasets, two captured in an underground station, and the publicly available MuHAVI dataset [123] and compared to manually labelled ground truth, as presented in Section 2.4.

The main contribution of this chapter is a framework for estimating accurately the pedestrian speed profiles of people walking unimpeded and with constant speed within a specific environment. For this purpose, the proposed framework attempts to track each pedestrian's step to pinpoint his/her locations over time and check the consistency of walking speed within the defined environment.

2.2 Previous Work

Estimation of the average ground plane speed of pedestrians who move in a defined space requires the start and end point of their movement along with the time needed to cover the distance between them. Since the speed of people is measured as the distance they cover on the ground plane divided by the interval of time needed, the most crucial aspect for estimating the speed of pedestrians is to obtain a sequence of their tracked positions and translate them into real world coordinates. Although two points are sufficient to estimate the average speed of people, a sequence of measurements along their path can be used to assess the quality of the tracking and localizing procedure. Moreover measurements along the trajectory of a person can identify sudden changes in speed, due to changes of direction of movement or the presence of obstacles. Thus tracking a pedestrian throughout a frame sequence is essential in order to confirm that the pedestrian sustains a constant walking speed and thus walks unimpeded.

While the position of a pedestrian may be specified by the body centre of gravity [89] or the top of the head [120], converting the location coordinates to ground plane coordinates may be inaccurate, unless the height of the pedestrian is explicitly known. Thus tracking the lower part of the foot positions of a pedestrian is an attractive option [88], as it allows direct association between tracked pedestrian locations and their correspondence to the ground plane.

Pedestrian tracking may be based on motion segmentation, pedestrian appearance detection or local feature correspondence. Motion segmentation aims to detect pedestrians as moving regions in image sequences. Motion segmentation techniques include temporal differencing, optical flow and background subtraction. For a survey on motion segmentation techniques the reader is referred to [10][112]. In addition, ChangeDetection.net (CDNET)[50] , maintains an online record and comparison of motion segmentation algorithms.

In temporal differencing [84], moving regions are detected by pixel-wise differences between two or three consecutive frames. Similar to temporal differencing, in change vector analysis (CVA) [22] each image pixel is represented with a multispectral feature vector and the difference image is estimated by calculating, for each pixel, the modulus of the difference between two feature vectors. In optical flow techniques [91] the flow of

vectors of moving objects is being used in order to perform motion segmentation. In background subtraction techniques, motion is detected as the difference between the current image and the reference background, as in [70] where an adaptive mixture of Gaussians is used to model the background. Although motion segmentation algorithms may detect isolated pedestrians, they tend to fail to distinguish between pedestrians that occlude each other.

Alternatively, pedestrians may be detected using pedestrian appearance models, such as boosted *edgelets* body part detectors [137], Histograms of Oriented Gradients – HOG [33] that model the appearance of a person using statistical models, mixtures of multiscale deformable part models [42] and part based detectors which are tightly clustered in both appearance space and in body configuration such as *poselets*[18]. These algorithms overcome the limitation of the segmentation methods by differentiating between individuals, even when they are close to each other. However they tend to be very sensitive, producing many false positives and their localization accuracy is not sufficient for precise tracking.

Pedestrian detections (blobs) extracted by either motion segmentation or appearance model methods are normally temporally grouped into trajectories using methods such as the *Kalman* filter [71] or the particle filter [66] that associate detections across consecutive frames.

In feature-based tracking, a set of interest points of sufficient saliency is selected and matched in successive frames and then spatiotemporally clustered into pedestrian trajectories. For instance, Kanade-Lucas-Tomasi – KLT features are used in [110] while in [107] Harris corners as features are employed. Although both approaches are able to detect and track walking pedestrians, the spatial accuracy of the tracked individuals is not sufficient for speed measurement.

Accurate speed estimation requires pinpoint accuracy of the pedestrian location on the ground plane. Such an accurate estimation may be the result of tracking of a specific point of the human body (e.g. top of head, middle of torso, lower pelvis, heel). However, if this point is not on the ground plane, its projection on the ground plane will be ambiguous, even if camera calibration is used, because of the uncertainty of the pedestrian height.

While attempts have been made to automate the collection of pedestrian data to estimate the preferred walking speed, the resulted precision is lower than what is required in pedestrian simulation software. Ismail, et al. [67] presented a system for automated pedestrian speed estimation. Camera calibration is performed by collecting linear field-of-view observations of entities appearing in the video images and the track of pedestrians calculated using the KLT feature tracker. In [63] a temporal differencing technique is applied for motion segmentation. Then the individuals are tracked and their image positions translated to real world coordinates, using a model to compensate for the pincushion distortion. Finally a Kalman filter is being applied for post processing the estimated velocities. Both approaches are applied on far-away top-down view scenarios, where the pedestrians are small in size and their ground plane locations can be easily specified. However, small errors in image-based position estimation may have a significant impact on the speed estimation and cause significant errors. Thus in different scenarios such as looking at pedestrians sideways from a closer distance, their approach may not be able to provide accurate results. In [61] a semi-automatic system for pedestrian speed estimation is presented. The heads of pedestrians are tracked and their speed estimated in an observation area which has been measured using a calibration stick. However the camera view is not vertical to the floor and thus the accuracy of the speed estimation is questionable due to the variation in the height of people.

2.3 Speed Estimation

In this section, two systems are presented for pedestrian speed estimation, the Heel Localisation System (HLS) and the Step Localisation System (SLS). Although each system deals with different scenario, they share an architecture with similar parts. In the first scenario, pedestrians walk on flat ground perpendicular to the camera plane and are viewed from the side. In the second scenario, pedestrians walk down stairs and are viewed from the front.

We propose that speed estimation accuracy of walking pedestrians may be improved by foot localisation at midstance phases (i.e. static foot) during the walking cycle, where a pedestrian's foot touches flat the ground. This approach is adopted in both systems with variations on foot localization and combined with foreground-background modelling and

camera calibration. For both systems, initially motion detection is applied through foreground-background separation and the detected foreground blob is tracked. Then, the static foot of the pedestrians gait is located. Finally, their ground plane speed is calculated, by projecting the spatiotemporal information of the pedestrian feet, using a camera calibration model. An overview of the systems methodology is presented in Figure 2.1 In Section 2.4, comparative results are presented to demonstrate how our proposed systems improve the accuracy of speed estimation, compared to standard blob tracking.

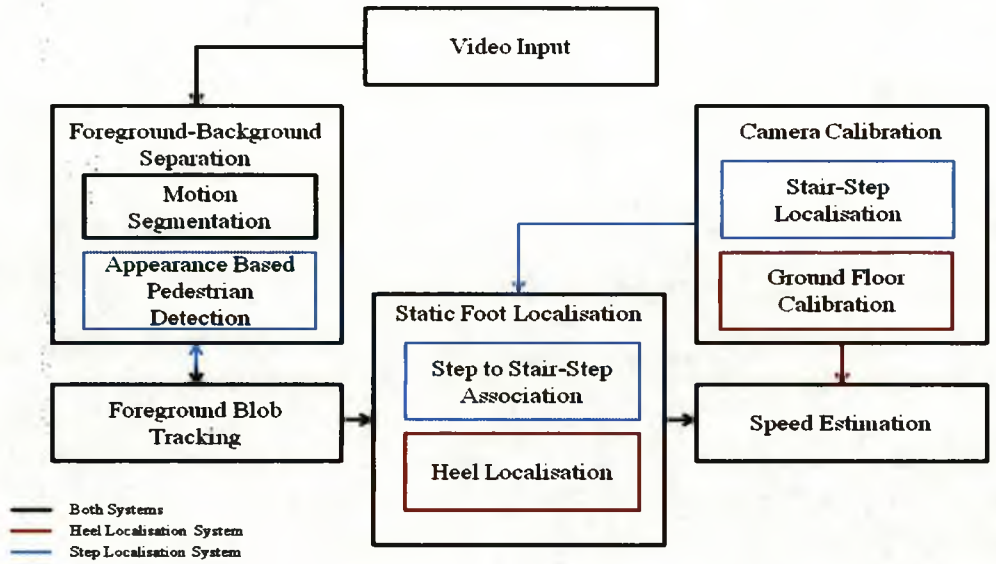


Figure 2.1: Proposed Methodology

While motion segmentation may give satisfactory results for motion parallel to the camera plane (HLS), self-occlusions cause problems when motion is perpendicular to the camera plane (SLS). Therefore, a foreground enhancement approach based on appearance-based pedestrian detection is applied for the latter case.

Static foot localization for the HLS is performed by estimating the spatiotemporal information of the heel of the static foot. However, this approach may not be applicable when pedestrians move vertically to the camera plane. Therefore, in SLS the blob position over time is estimated and associated with a step on the stairs.

In the HLS pipeline, a camera calibration model is being used to project the locations of the heels identified into real world coordinates. In the SLS pipeline, the steps of the stairs are manually localized and associated with real world coordinates.

2.3.1 Foreground Blob Estimation and Tracking

Let assume the input frame I_t at frame t. The binary foreground image G_t , with size same as I_t, is obtained using the approach of [70]. In the foreground image, components that are connected together are identified as single blobs and a mean shift particle filter (MSPF) [28][97] is applied to track these blobs through the sequence of frames. Both motion detection and multi-target tracking methods are implemented in *OpenCV* [19]. For a tracked foreground blob i, the predicted bounding box that a blob will occupy at time t is $B_{t,i}^{T}$. Each bounding box $B = [\mathrm{x}, \mathrm{y}, \mathrm{w}, \mathrm{h}]$ is defined by the coordinates of its top left corner (x, y) , its width w and height h.

In the SLS pipeline, the prediction derived by the tracking filter (MSPF) is combined with an appearance-based person detection algorithm (HOG [33]) to enhance the quality of the foreground. Such enhancement is necessary since the presence of self-occlusions results in mislabelling of parts of the human body as background.

By applying the HOG detector at frame I_t, j bounding boxes $B_{t,j}^{H}$ $(j \geq 0)$ are obtained. It is assumed that in the projection of a bounding box on G_t, either predicted from the tracker or the HOG detector, the foreground information should represent a single human subject. Thus by trying either to close foreground holes, or connect separate blobs in these projected boxes, we obtain a better representation of where pedestrians are, in order to localise them and feed this information to the tracker to estimate the position of a pedestrian for the following frame.

A foreground enhancement area $B_{t,i}^{E}$ of the bounding box $B_{t,i}^{T}$ is defined in Equation 2.1, depending on the overlap of $B_{t,i}^{T}$ and the HOG-detected bounding boxes $B_{t,j}^{H}$ in frame I_t . Specifically, $a_o(t, i, j)$, which measures the relative overlap of the two blobs, is calculated according to Equation 2.2. If there is no response from the HOG detector or there is no overlap between the two boxes, the enhancement area is defined by $B_{t,i}^{T}$ only. If there is

some overlap but a_o is lower than T_α, then the enhancement area is defined by the union of the two bounding boxes. $B^H_{t,j}$ is solely selected as enhancement area when there is sufficiently high overlap between $B^T_{t,i}$ and $B^H_{t,j}$ (a_o>T_α) in order to reduce the foreground area to be enhanced, minimizing thus any noise enhancement. In this latter case, $B^H_{t,j}$ is preferred over $B^T_{t,i}$ because the tracker may be more prone to prediction errors (i.e. position of predicted tracking bounding box) due to noise in the foreground mask it uses to predict future positions of tracked blobs. The value of T_α is found empirically as it is environmental specific and depends on the resolution of the frame and the size of people, in pixels, appearing in it.

$$B^E_{t,i} = \begin{cases} B^T_{t,i} \ , \ if\ a_o(\mathrm{t,i,j}) = 0 \\ B^T_{t,i} \cup B^H_{t,j} \ , if\ \ 0 < a_o(\mathrm{t,i,j}) \leq T_a \\ B^H_{t,j} \ , if\ \ a_o(\mathrm{t,i,j}) > T_a \end{cases} \qquad (2.1)$$

$$a_o(\mathrm{t,i,j}) = \frac{area(B^T_{t,i} \cap B^H_{t,j})}{area(B^T_{t,i} \cup B^H_{t,j})} \qquad (2.2)$$

Foreground enhancement may be achieved by following two approaches. Both approaches aim to connect foreground components that belong to the same person, but seem separated in foreground image. The first approach uses a morphological closing operator in order to fill gaps in the foreground of $B^E_{t,i}$. In the second approach, all foreground blobs within $B^E_{t,i}$ are connected so to appear as one unified blob. Figure 2.2 shows examples of the foreground enhancement algorithms. Although the quality of the foreground blobs extracted by the two approaches may differ significantly, e.g. the blob after closing (Figure 2.2 (b)) seems more accurate than the blob after connecting separated regions (Figure 2.2 (c)), both of them will result to similar outputs, as SLS considers the lowest mid-point of the tracked blob, as discussed later in Section 2.4.

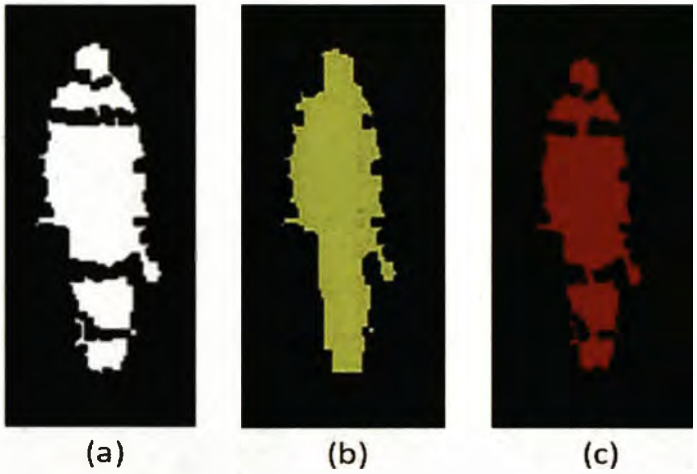


Figure 2.2 : (a) Original foreground pixels. (b) Blob after morphological closing. (c) Blob after connecting separated regions.

2.3.2 Step Estimation

2.3.2.1 Heel Localisation

To identify the real world coordinates of pedestrians' positions, when they walk on a flat surface (ground plane), camera calibration is performed using Tsai's coplanar calibration method [132]. The extracted camera model allows direct conversion of the image-based coordinates to real world coordinates, as long as they are constrained on the ground plane.

Let assume that the only moving objects in the video sequences are pedestrians who walk parallel to the image plane I_t, so every blob $B_{t,i}^{E}$ is considered to correspond to a pedestrian. In our pedestrian simulation application, we are interested in modelling only pedestrians who walk unimpeded, and in straight lines in the real world environment, therefore speed measuring is constrained to blobs with constant speed, and with no change in direction, throughout the frames.

To measure the speed of a walking pedestrian the system needs to identify, localise and track a specific body part of this person throughout a video sequence. Pedestrian feet are appropriate, because ground plane positions can be derived directly, without any assumption about the height of the pedestrian, and thus allowing accurate speed estimations. Also, the speed of other body parts, such as head and torso, varies naturally

during the walking cycles due to the biomechanics of walking and checking the speed consistency on specific times within these cycles (e.g. midstance) addresses this issue.

In [17], a Harris corner detector is applied on a sequence of frames, to locate the feet of pedestrians, in order to analyse different patterns of walking gait and to classify a tracked blob, as a single person, a group of people or an undefined object. In [88], a similar approach is used to track the lower body parts through a video sequence. We further extend this method to allow accurate pinpointing of the heel of the foot location of a pedestrian.

When pedestrians are viewed from the side they appear in an upright position, with their feet located at the bottom of their body. Thus using the location (x,y) of the top left corner of the bounding box $B_{t,i}^{T}$ along with its width (w) and height (h), an area $L_i(t) = [x - {}^{w}/_{2}, y + {}^{3h}/_{4}, {}^{h}/_{2}, 2w]$, (Figure 2.3), is defined where the heel location is searched. The first two entries of $L_i(t)$ specify the coordinates of the top left location of the rectangular area while the last two specify its height and width.

Human gait is characterized by its cyclic nature, where there is a periodic movement of feet. A foot during walking alternates between two phases. The stance phase is where the foot is in contact with the ground and the swing phase is where the foot moves, not being in contact with the ground, towards the next foot strike. When fully able people walk, their feet are 62% of the time in the stance phase and this includes the time from the moment the hill strikes the ground till the moment the toes leave the ground for the swing phase [135]. Since a person is in constant motion during walking, feet are the only part of the body which remains static for a significant period of time. Thus the accumulation of image corners located in the area of a static foot must be higher, while corners corresponding to the moving foot or other body parts are spread in larger areas due to their motion. However, a limitation of this approach is that a high number of corner responses should be generated in order to provide clues regarding the foot location. Thus a person in a frame must appear in a rather large resolution else the accuracy of recognition will be decreased.

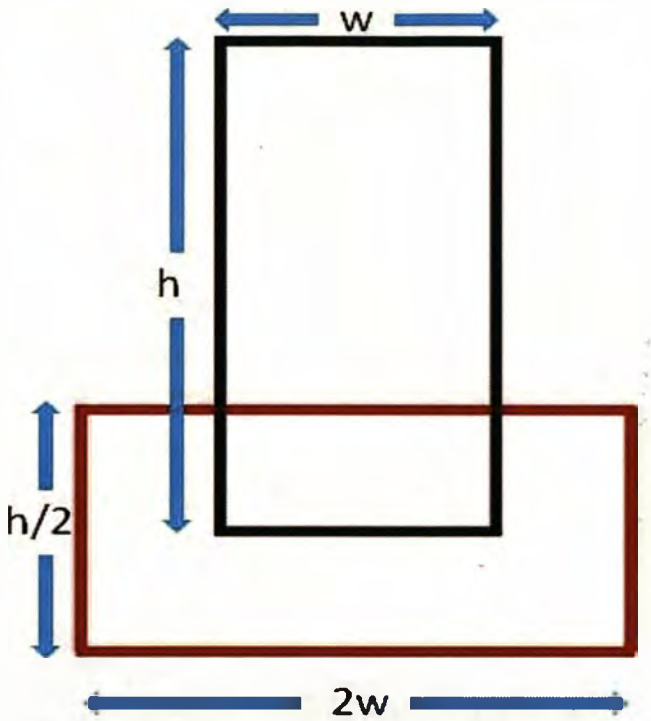


Figure 2.3: The black rectangle is the blob bounding box, while the red rectangle defines the heel search area $L_i(t)$

For every frame, a corner detection algorithm [119] (Figure 2.4(e)) is applied on the area $L_i(t)$(Figure 2.4(d)). For every tracked human *I*, corners $c^i_{x,y}(t)$ found at pixel (x,y) for every frame *t*, are then accumulated in a 2D histogram map C^i with the same size as the original image I_t.

$$C^i_{x,y} = \sum_t c^i_{x,y}(t)$$

$$c^i_{x,y}(t) = \begin{cases} 1, if\ corner\ in\ I^i_{x,y}\ at\ frame\ t \\ else\ 0 \end{cases} \quad (2.3)$$

Also, a set $T^i_{x,y}$ is created for each pixel *(x,y)* and for each tracked pedestrian *i*, which stores the temporal information of the presence of the corners. That is:

$$T^i_{x,y} = \{t: if\ c^i_{x,y}(t) = 1\} \quad (2.4)$$

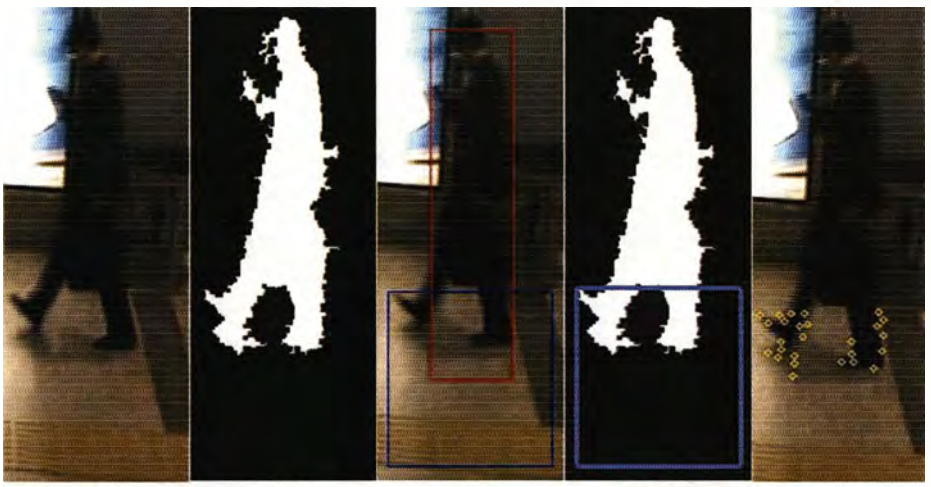


Figure 2.4: a) Part of the original frame, b) foreground pixels, c) bounding box $B_{t,l}^{tr}$ (red box) heel search area $L_l(t)$(blue box) d) heel search area $L_l(t)$(blue box), e) detected corners in the heel search area.

High values in the histogram map C^i are expected at the static foot locations of pedestrian i over time. However, it is difficult to identify local maxima in the histogram map due to its sparsity. Therefore, a corner 2D map P^i for each pedestrian i is calculated, by smoothing the histogram map, e.g. using a mean average or/and a *Gaussian* filter of size *NxN*. The result of this process is illustrated in Figure 2.6(a). Then, in order to enhance the intensity of values at the heel location the filter A^c with size *NxN* too, is used, where the anchor of the kernel lies in the centre of the filter:

$$A^c = \begin{cases} 1, if\ r < \Psi\ and\ (c < \Psi\ if\ the\ person\ is\ moving\ \ right\ to\ left, else\ c > \Psi) \\ -2, \quad otherwise \end{cases} \tag{2.5}$$

where $\Psi = \frac{N}{2} + 1$ and r and c represent the row and the column of the filter respectively.

Along with the filter of Equation 2.5, two other similar filters, were used in the experiments. First A^v (Equation 2.6) which enforces the peak location at the lower part of

the foot and secondly A^h(Equation 2.7) which enforces the peak location at the rear part of the foot.

$$A^v = \begin{cases} 1, if\ r < \Psi \\ -2, \quad otherwise \end{cases} \tag{2.6}$$

$$A^h = \begin{cases} 1, if\ c < \Psi \\ -2, \quad otherwise \end{cases} \tag{2.7}$$

The value of N is estimated as the average over time of the ratios of foreground pixels contained in the lower part $L_i(t)$ over the number of corners $c^i_{x,y}(t)$ in the same area, and it is always rounded to the closest odd integer. Therefore, as the number of corner responses increases, the size of the smoothing window decreases in order to preserve the distribution of the high accumulation corner areas. Otherwise, when the corner responses are fewer the size of the smoothing window increases in order to spread the number of corner votes in the local neighbourhood. Finally the local maximum (see Figure 2.6(b)) is located by scanning a window of size $(a * N)$x$(a * N)$, where a is parameter that controls the size of the scanning window.

1	1	1	1	-2	-2	-2
1	1	1	1	-2	-2	-2
1	1	1	1	-2	-2	-2
1	1	1	1	-2	-2	-2
-2	-2	-2	-2	-2	-2	-2
-2	-2	-2	-2	-2	-2	-2
-2	-2-	-2	-2	-2	-2	-2

-2	-2	-2	1	1	1	1
-2	-2	-2	1	1	1	1
-2	-2	-2	1	1	1	1
-2	-2	-2	1	1	1	1
-2	-2	-2	-2	-2	-2	-2
-2	-2	-2	-2	-2	-2	-2
-2	-2-	-2	-2	-2	-2	-2

Figure 2.5: Filters generated using Equation 2.5 for N=7. The red cell indicates the position of the kernel anchor. The left filter is applied when a person is moving from right to left, while the right when a person is walking to the opposite direction.

The corner responses outside the foot area are suppressed and at the same time the corner responses in the heel neighbourhood are enhanced to facilitate the estimation of the heel location. The filter in Equation 2.5 is based on the assumption that in the neighbourhood of the heel, below and opposite the direction of the pedestrian movement the accumulation of the corner responses would be minimal. In Figure 2.5 two filters for N=7 are shown, one for each walking direction.

The set of peaks (local maxima points) contains estimates of the heel positions along with potential outliers. Assuming that a person is walking straight, which is consistent to our requirement that pedestrians walk unimpeded, the peaks which correspond to the heel touching the ground must be located on a straight line. Therefore a line is fitted on the peak locations, using the Least Median of Squares method [65], and any outlier peaks whose distance from the line is above a threshold ϑ_α are discarded (see Figure 2.6(c)).

The set of the remaining peaks may contain multiple estimates of the same foot (e.g. because of high concentration of corners in the front and the back of the foot). This is because their distance (foot size) may be larger than the size of the scanning window used to search for local maximum. The temporal information of corners as recorded in $T^i_{x,y}$ is used in order to calculate the average frame φ_k at which a corner was detected for step k. Peaks with a temporal difference of average frames of appearance lower than a threshold θ_β are considered to belong to the same step and thus are grouped together in a set S_k, where $k = 1,2,..,N_f$ and N_f is the total number of steps (see Figure 2.6(d)). Since each step of a pedestrian should be described by only one peak, all the possible combinations of peaks Q_ω are derived, where:

$$\omega = 1..N_\omega \; and \; N_\omega = \prod_{k} |S_k| \qquad (2.8)$$

To identify which combination describes best the walk of the pedestrian under examination, the constraint of constant speed is applied, or equivalently the notion that all steps should have similar lengths. Therefore, the combination of heel points that leads to the smallest variance in step lengths is selected(see Figure 2.6(e)). Figure 2.6 summarises all the stages of heel localization procedure using one sequence. The corner map in Figure 2.6(a) has been enhanced by applying the horizontal filter A^h (Equation 2.7).

Figure 2.7 shows some examples of foot localization by applying different filters. As displayed in (Figure 2.7(a)), no filtering results in a peak around the centre of distribution of corner responses. In Figure 2.7(b), A^h (Equation 2.7) has been applied, which forces the peak location along the back side of the foot. In Figure 2.7(c), A^v (Equation 2.6) is used which results to a peak along the lower part of the foot. Finally in Figure 2.7(d), the use of

filter A^c (Equation 2.5) forces the detection of the peak to be at the heel location of the foot. Therefore, filter A^c is preferred, as it provides pinpoint estimation of heel locations.

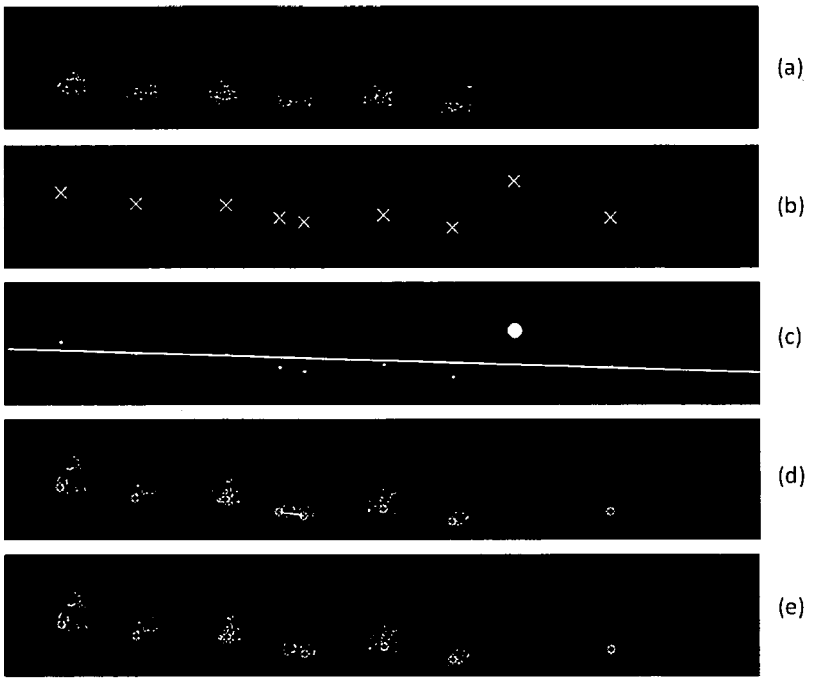


Figure 2.6: a) Corner response map image P^l, b) local maxima of corner neighbrhoods, c) discarding any outliers (large circle) that do not fit the assumption of straight walking, d) local maxima that seem to belong to the same foot are linked together (white line is for illustration to show maxima that are identified belonging to the same foot), e) final feet location estimations by minimizing the variance on step length.

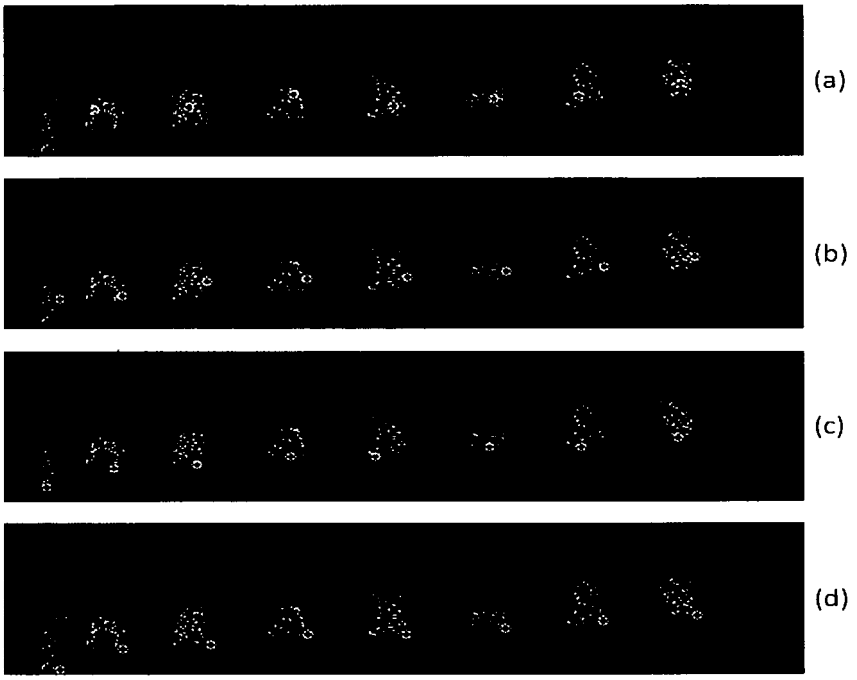


Figure 2.7 : Comparison of filters applied to the Corner map image of a pedestrian walking from right to left. Circles indicate detected heel location. a) No filter b) Horizontal filter A^h (Equation 2.7) c) Vertical Filter A^v (Equation 2.6) d) Proposed FilterA^c (Equation 2.5).

2.3.2.2 Step to Stair-step Association

The problem of estimating the speed of pedestrians walking up/down a staircase seen by a frontal camera is treated as a problem of measuring the speed along the horizontal dimension, as seen in Figure 2.8. Let assume a Cartesian co-ordinate system such as the width and the height of each step correspond to the x-axis and the y-axis respectively. The speed of pedestrians is actually measured on the projection on x-axis. Assuming that the width w_s of each staircase step s is known, the whole x-distance l that a pedestrian will cover while walking on the stairs is equivalently known as:

$$l = \sum_{s=1}^{N_s} w_s \quad (2.9)$$

where N_s is the total number of steps of the stairs.

The steps are manually localised as lines which are located at the end of each physical step. The end of each stair-step is used because it is easy to be identified. In Figure 2.8(b) a frame of the video input is displayed with the position of each step being marked with a red line.

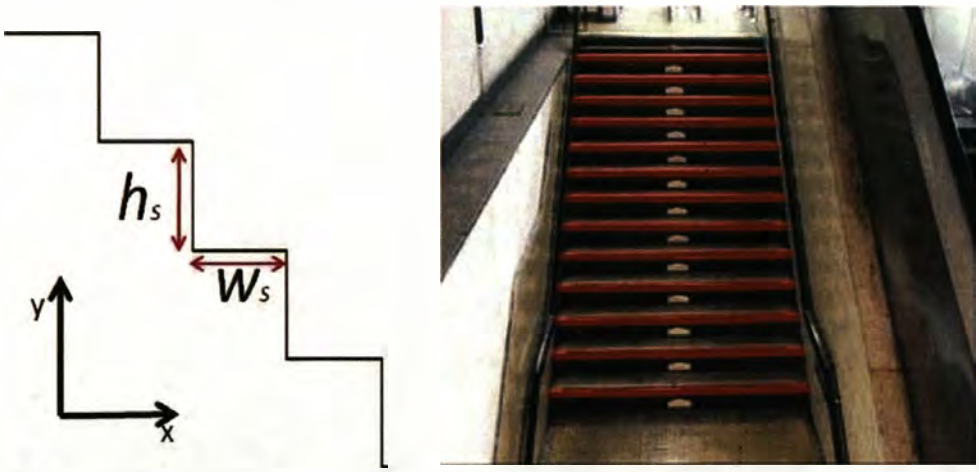


Figure 2.8 : Step localization, a) graphical representation of stairs defining the x, y axis and width and height of each step, b) each artificially made red line represents a step of the stair.

For each tracked person i, the spatiotemporal information of its tracked path is recorded. This information consists of the spatial position of the lower part of the estimated (updated prediction) tracking bounding box $B_{t,i}^{T}$ (Figure 2.9) along with the frame number t. The lowest midpoint $p_{t,i}$ is chosen, since it is a good indicator of the position of the feet of the pedestrian, where:

$$p_{t,i} = \begin{bmatrix} x_{t,i}^{T} + {w_{t,i}^{T}}/{2} \\ y_{t,i}^{T} + h_{t,i}^{T} \end{bmatrix} \quad (2.10)$$

Figure 2.9: The position of the lowest midpoint (yellow circle) of the tracked pedestrian is recorded.

However, position $p_{t,i}$ may be inaccurate because of tracking errors. In order to discard these inaccurate points, the locations of the points recorded for the next λ frames after frame t are considered. Specifically the variance of the differences of the vertical positions of succeeding points is examined and if this variance is below a threshold T_d then the point is accepted for further processing. That is:

$$p_{t,i} \text{ is accepted if } \sigma(d) < \mathrm{T}_d \text{, where}$$
$$d_{t,i} = y_{\tau,i} - y_{\tau+1,i}, for\ \tau \leq t + \lambda \quad (2.11)$$

The next step is to associate each accepted point $p_{t,i}$ with one of the M stair-steps and assign it to a correspondent set $S_{s,i}$ based on its distance from each step on the image plane. Only points that are sufficiently close to the step lines are taken in consideration. So:

$$p_{t,i} \in S_{j,i}\ if\ arg\min_s(D_{p,s}) = j \text{ and } D_{p,s} < \mathrm{T}_e \quad (2.12)$$

where $D_{p,s}$ is the distance on the image plane of $p_{t,i}$ from the line that defines stair-step s and T_e is a distance threshold.

2.3.3 Speed Estimation

2.3.3.1 Speed Estimation in Heel Localisation System

The speed V_i^k for each step k of a pedestrian i is estimated based on the static foot locations by the formula below

$$V_i^k = \frac{r \cdot d(R^k, R^{k+1})}{\varphi_{k+1} - \varphi_k} \qquad (2.13)$$

where R^k is real world coordinate of the heel location for step k, φ_k is the average frame of corner detections that were associated with step k, as explained in Section 2.3.2.1 and r is the frame rate of the video sequence. Similarly the average speed $\overline{V_i}$ of a pedestrian i is estimated by considering the first and last static foot locations detected.

One of the requirements is to identify and measure the speed for only the pedestrians that walk with constant speed. To identify these pedestrians the mean step speed $\overline{V_i}$ for each pedestrian i is calculated and then considered valid if it satisfies the following equation, where T_f is a threshold value.

$$\max_k \left(\frac{|V_i^k - \overline{V_i}|}{\overline{V_i}} \right) < T_f \qquad (2.14)$$

2.3.3.2 Speed Estimation in Step Localisation System

After associating each point $p_{t,i}$ with a stair-step s (i.e. $p_{t,i} \in S_{s,i}$), an iterative procedure takes place in order to eliminate points resulted from noisy measurements. Therefore the mean frame $\overline{\Gamma_{s,i}}$ of the frames that belong to set $\Gamma_{s,i}$ is calculated, where $\Gamma_{s,i}$ is a set of frames of all the points that belong to $S_{s,i}$. Then, any points from $S_{s,i}$, whose frames are more than one standard deviation $\sigma(\Gamma_{s,i})$ away from the mean frame are discarded and $\Gamma_{s,i}$ is recalculated. This is an iterative procedure and it continues until no point is discarded.

Sets with less than 3 members are discarded as they are not considered to have sufficient support. The algorithm for discarding the points is summarised in Figure 2.10.

1) $t \in \Gamma_{s,i}$ if $p_{t,i} \in S_{s,i}$
2) Calculate $\overline{\Gamma_{s,i}}$ and $\sigma(\Gamma_{s,i})$
3) Discard all $p_{t,i}$ from $S_{s,i}$ with $\text{abs}(\overline{\Gamma_{s,i}} - \text{t}) > \sigma(\Gamma_{s,i})$
4) If number of $p_{t,i}$ discarded > 0 goto 1
5) Discard $S_{s,i}$ if $|S_{s,i}| < 3$

Figure 2.10: Algorithm for discarding noisy points and non popular sets

Then, the speed V_i of a pedestrian may be estimated, by considering the final $\overline{\Gamma_{s,i}}$ for, the first s_f and last s_l steps which a pedestrian crossed, the width w_s of a step and the frame rate f of the video sequence:

$$V_{i=} \frac{(s_l - s_f) \cdot w_s \cdot f}{\overline{\Gamma_{s_l,i}} - \overline{\Gamma_{s_f,i}}} \quad (2.15)$$

2.4 Results

2.4.1 Evaluation of Heel Localisation System

Two datasets are used to evaluate HLS: the HongKong-sideview dataset and the publicly available MuHaVi dataset. The HongKong-sideview dataset consists of 6 video sequences (720x576 pixel resolution) of pedestrians recorded in an underground station in Hong Kong (a representative frame is displayed in Figure 2.11). The duration of the combined video footage is one hour and presents various challenges, such as varying illumination, background motion and partial or total occlusions between pedestrians. We are interested

measuring the speeds of people walking in the foreground as seen in Figure 2.11. People moving in the background, generate 'false' detections which makes tracking even harder.

The dimensions of the floor tiles are known and an artificial board is constructed (see Figure 2.12) based on the tiles position. Therefore, the corner locations of the checkerboard are known both in the 2D image coordinates and in 3D real world coordinates and therefore the camera model may be estimated, using the Tsai calibration method [132]. The various parameters discussed in Section 3 were empirically selected and the results presented are using the best possible configuration.

In order to produce the ground truth data the position of the heel strike for each step of each pedestrian under consideration is manually marked at the frame when the foot becomes static and inside the measurement area. In total, 724 pedestrians which moved on a straight line are marked and their speed for each step is calculated. Out of those, 17 were discarded as they didn't satisfy the constant speed constraint.

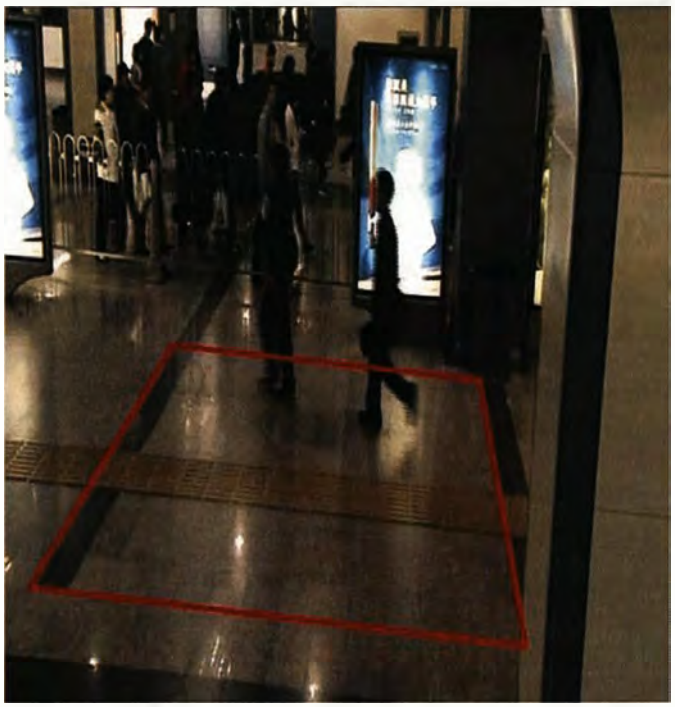


Figure 2.11: Sample frame from the HongKong-sideview video sequence. The red line encloses the area within which measurements were estimated.

Figure 2.12: An artificially-made check board on top of tiles is used to perform camera calibration

For the HLS and the MSPF tracker, 502 pedestrians were tracked. From these, those with two or less footsteps detected were discarded (104 people), since this implies that tracking was not sufficiently reliable (five to eight steps are normally needed to cover the distance from the right side to the left side of the enclosed area in Figure 2.11) to extract sufficient information. Moreover another 145 pedestrians did not satisfy the constant speed constraint (63 for MSPF) and they were discarded from the measurements, leaving 253 people for the HLS and 335 for the tracker, to compare against the ground truth data.

Discarding some pedestrians is not an issue for our application, because the purpose of the system is to generate a speed profile of pedestrians. On the other hand, it is important that the pedestrians whose speed is estimated are being tracked reliably and their speed is being estimated with high accuracy. Therefore any tracked pedestrian whose speed estimation might not be accurate is discarded.

To demonstrate the value of foot heel localization in the accuracy of pedestrians' speed, we compare the results obtained by HLS with the speed estimates calculated from only the bounding box tracking. For fair comparison, the results of bounding-box tracking are

converted to a sequence of steps. This is achieved by sampling uniformly the trajectory of the mid-lower point of the bounding box between the frames of the first and last step as identified by the foot localization system. After discarding the same 104 pedestrians which were discarded from the HLS due to lack of information, a further 63 were discarded for failing the constant speed constraint. Thus, leaving 335 pedestrians from which the speed profile of the tracker was computed.

In Figure 2.13 the error rate of the two approaches is displayed. Using the foot localization system 69% percent of all measured pedestrians have less than 5% speed error, from the ground truth measurements, while 97% of all pedestrians have less than 10% error. On the other hand using only the bounding box tracker without foot heel localisation, at the same error rates the results are 50% and 93% of the pedestrians respectively.

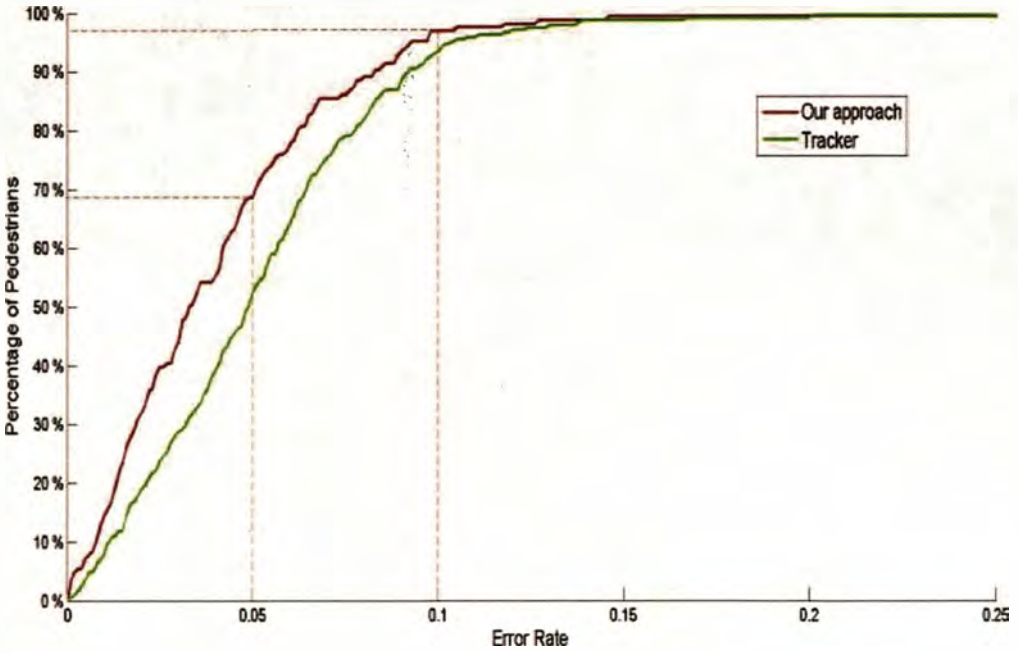


Figure 2.13: Cumulative distribution function of speed error rates for HLS.

Figure 2.14 displays the estimated speed profiles. The ground truth speed profile was constructed by taking into account all the 707 pedestrians manually marked. It can be observer that the speed profile distribution generated using the HLS is very similar to the

speed profile resulted from the ground truth data. The Bhattacharyya distance between the speed profile distribution of the ground truth and the foot localization system is 0.0963 while the Bhattacharyya distance of the ground truth from the bounding box tracker is 0.1686.

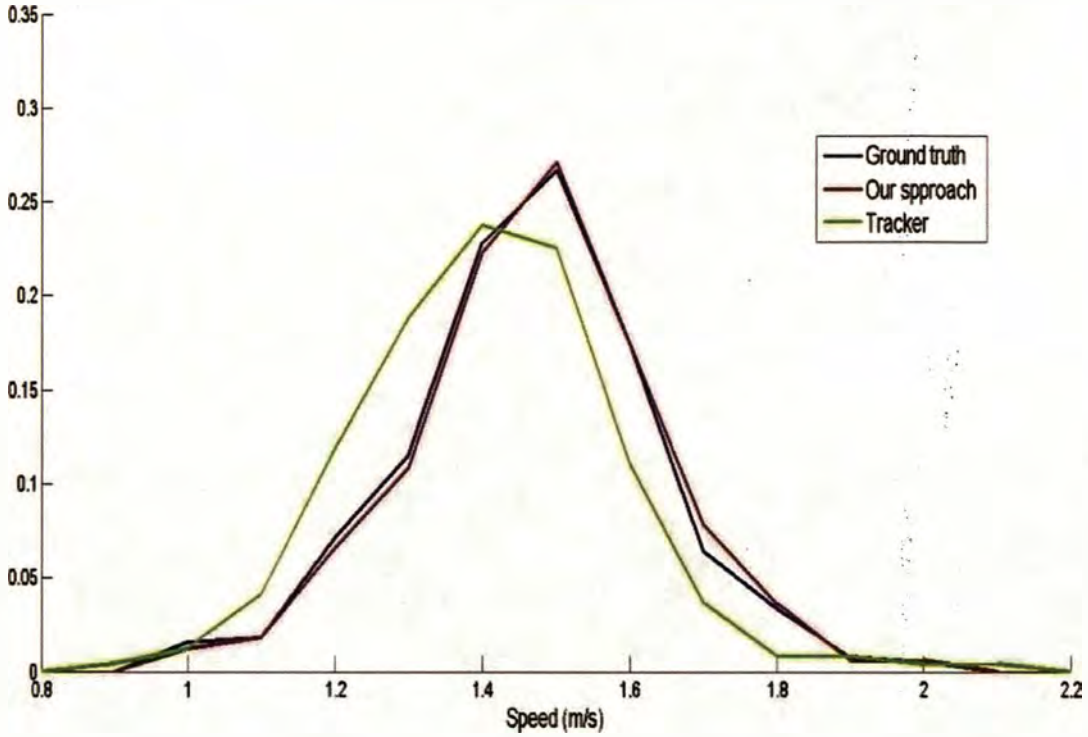


Figure 2.14: Speed Profile probability distributions for HLS.

We also validate HLS using a public dataset, the Walk Turn Back sequence of the MuHAVI dataset [123], (see Figure 2.15). Specifically the heel position for each step of the 4 runs, of each one of the 7 actors, is being estimated and the distance from the manual annotated heel strike is being measured.

Figure 2.15: Sample frame from MuHAVI dataset

In Figure 2.16 the distance between the estimated heel-strikes and the ground truth ones can be seen. 90% of all estimated heel strikes are less than 12cm away from the ground truth ones.

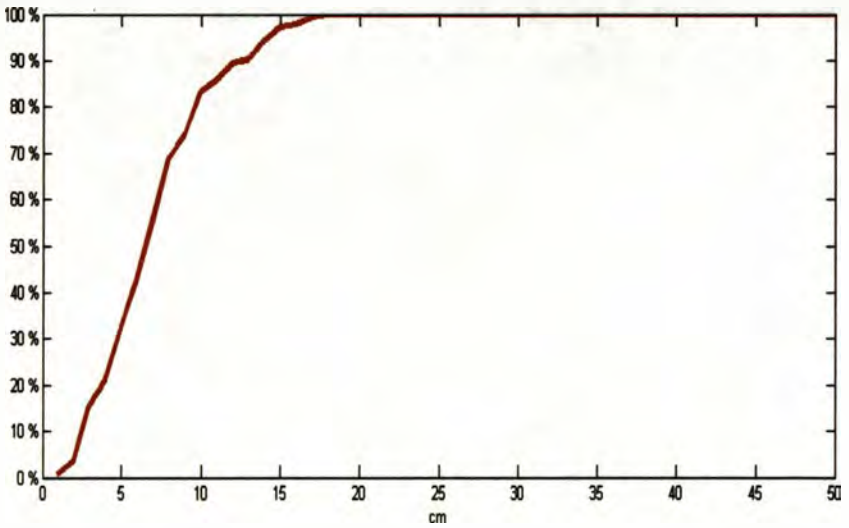


Figure 2.16: Distance from ground truth heal strike. Y axis shows the percentage of heel strikes.

2.4.2 Step Localisation System

The HongKong-staircase dataset is used to evaluate the SLS, which consists of three video sequences of people walking downstairs in an underground station in Hong Kong (see Figure 2.17). Unfortunately, no appropriate public dataset representing a staircase scenario exists for evaluating SLS. The resolution of the HongKong-staircase video sequences is 360x288 pixels and the total footage time is 30 minutes. The main challenges are the blend of the colour information of the pedestrians with the colour of the steps. Also due to the particulars of the movement of pedestrians on stairs, pedestrians may occlude each other and also pedestrians walking behind each other may be co-detected as big foreground blobs. Similarly to the previous scenario, the ground truth is obtained by manually marking the footsteps of pedestrians the moment their foot touches a step on the stairs. 193 pedestrians were marked, who walked down the stairs, stepping on all stair-steps and refrained from stopping in between them.

Different configurations of the system with or without the foreground enhancement algorithms (i.e. morphological closing operator, connected blobs) are compared. A 5-column by 15-row rectangular kernel is being used for the closing operation to overcome the problem of foreground background separation algorithm which misclassifies pedestrian's parts as background due to the same colour information with the back of the step in the y-axis of the stairs. The kernel is chosen such as to have similar aspect ratio to pedestrian foreground blobs.

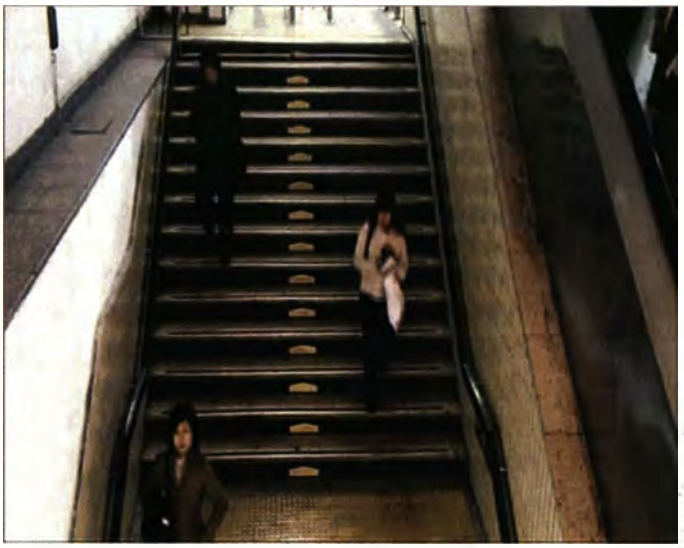


Figure 2.17: Sample frame from the HongKong-staircase video sequence, filmed in an underground station in Hong Kong.

From 193 marked pedestrians, the pipeline with the connected blobs algorithm discarded 84 as invalid (i.e. having information for less than 3 steps) while using the morphological operator 92 were discarded. Without the use of any of the foreground enhancement algorithms 105 pedestrians were discarded. As mentioned previously, the main objective is not measuring the speed of all pedestrians but measuring the speed of some pedestrians as accurate as possible.

In Figure 2.18, the error rate of the three different approaches can be seen. The blob tracker without any foreground enhancement achieved a 77% acceptance rate (pedestrian speeds with less than 10% error). The approach that applied morphological operations achieved around 82% acceptance rate, while the approach that applied connected components achieved 88% acceptance rate. Figure 2.19 shows the generated speed profiles. The blob tracker performed the worst with a Bhattacharyya distance of 0.1042. The morphological approach performed better with distance of 0.0875, while the connected blobs approach performed the best, generating almost an identical speed profile with that of the ground truth, and with a Bhattacharyya distance of 0.0346. The behaviour of both foreground enhancement methods is very similar; the fact that the results are better for the connected blobs methods is because in some cases the morphological operator is not able to connect

blobs that belong to the same person or it adds information to the foreground image that is not valid (i.e. it creates foreground information where it does not exist) and thus the tracking is inaccurate. On the other hand the connected blobs approach is prone to error when pedestrians walk close to each other. In cases where people are close and their bounding boxes, either generated from the tracker or from the HOG detector, overlap each other, it connects all the blobs together and thus the tracking is inaccurate.

Figure 2.20 displays an example of how the foreground enhancement algorithms work. In Figure 2.20(a) the original foreground information can be seen while in Figure 2.20(b) the foreground is enhanced using the connected blobs algorithm. In Figure 2.20(c) the foreground is enhanced using the morphological closing operator.

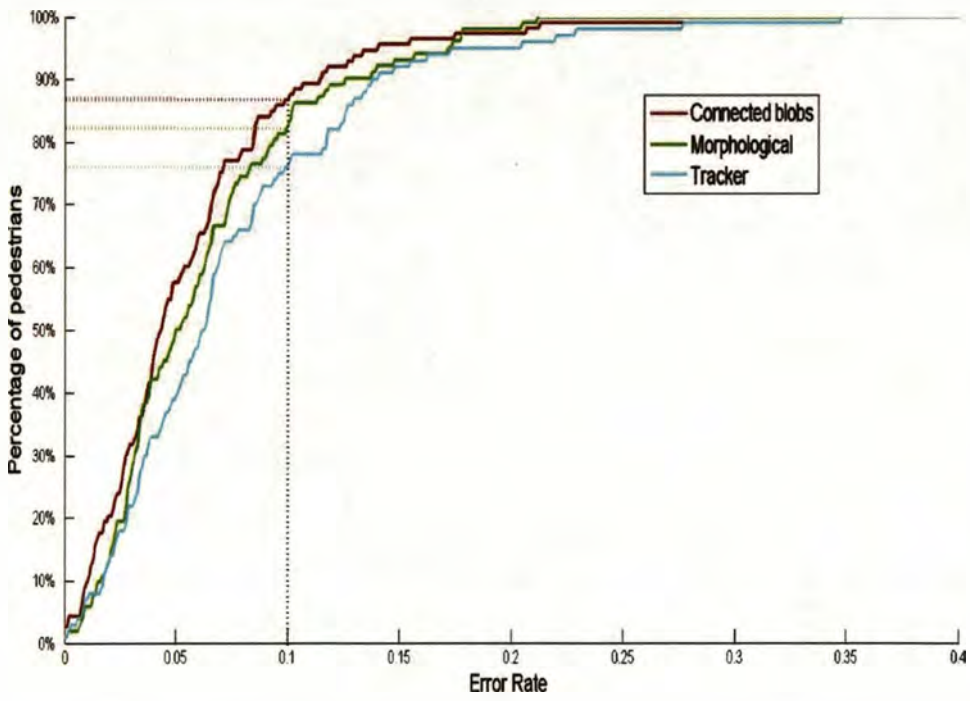


Figure 2.18: Cumulative distribution function of speed error rates for the SLS.

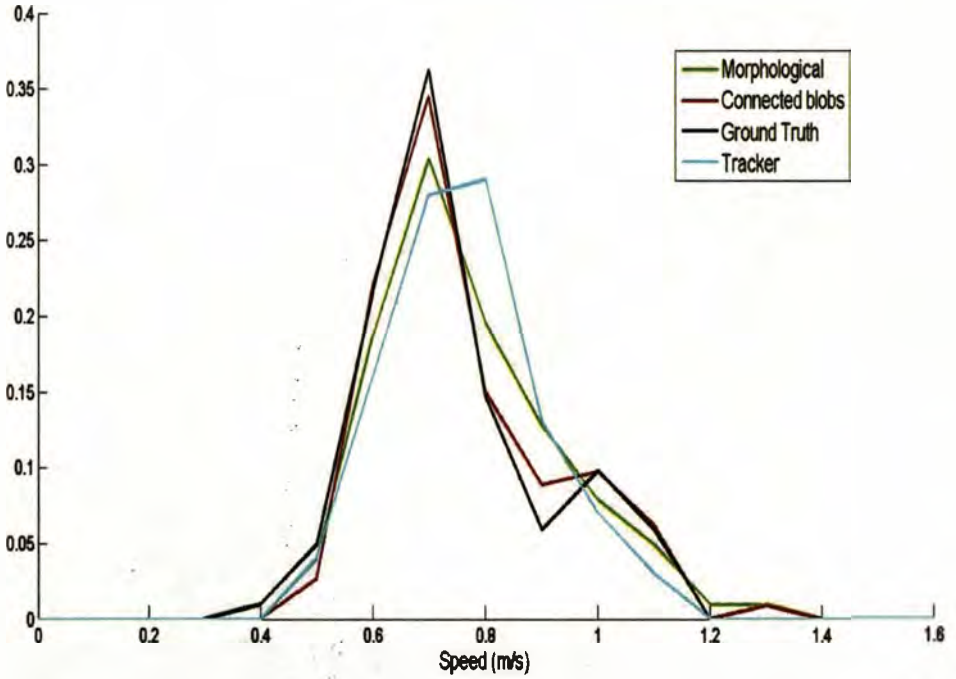


Figure 2.19: Speed profile probability didtributions for SLS.

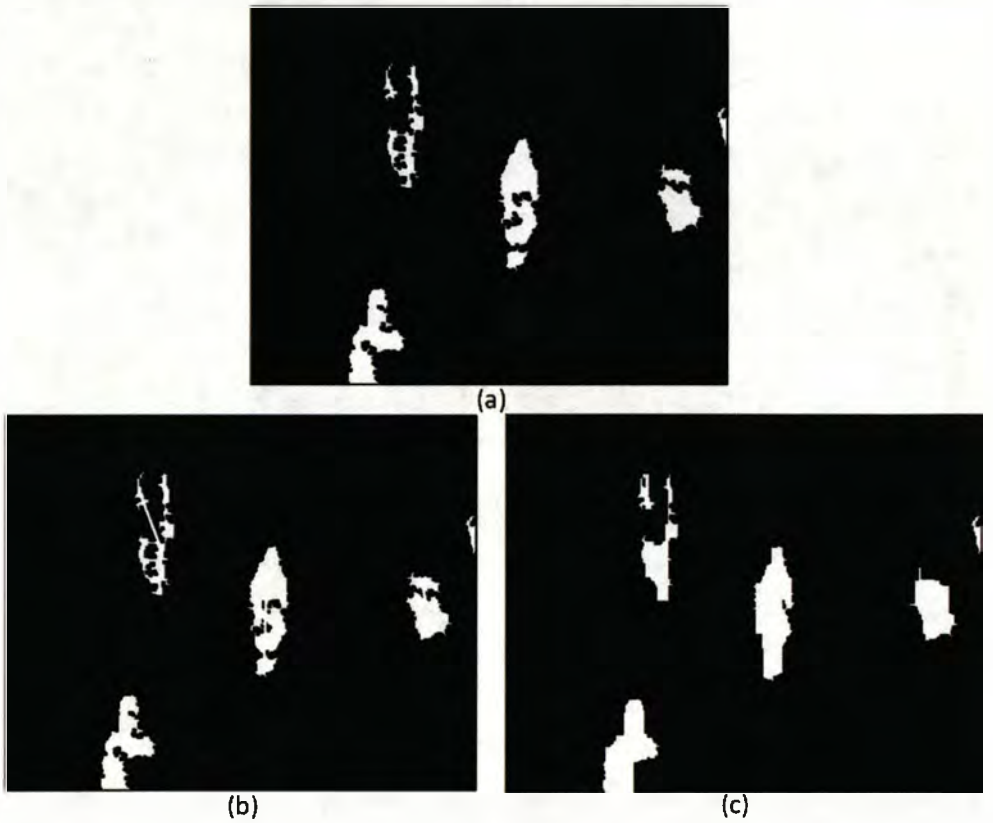


Figure 2.20: (a) original foreground, (b) foreground after connected blobs algorithm, (c) foreground after morphological closing.

2.5 Conclusion

The main contribution of this chapter is the development of a framework for pedestrian speed estimation. To my knowledge there is no system available that can estimate with such accuracy the speed profile distribution of pedestrians. Although some of the methods used for the foreground/background separation and the human detection might not be the state of the art, they may be considered as exchangeable modules to the framework and more recent methods such as in [14][87] and [126] for foreground detection, and methods for pedestrian detection such as those described in **Error! Reference source not found.** and [48] can be easily incorporated in the pipeline and substitute the methods used for this research.

In this chapter, it was also demonstrated that spatio-temporal static foot localisation may improve the accuracy of speed estimation, compared to standard pedestrian tracking approaches. Specifically, two systems that implemented different foot localization methods were presented to address two scenarios: Pedestrians walking sideways to camera view and pedestrians walking down the stairs while viewed from the front.

In the side view walking scenario, the results have shown that the estimated speed profile is highly accurate with a Bhattacharyya distance of 0.0963 from the ground truth speed profile distribution and with an acceptance rate of 97%. The heel localization system may fail to track successfully some individuals, because of noise around the heel strike positions. Fortunately, such erroneous tracks are filtered out by the constant speed assumption. In the staircase scenario, although we achieved 88% acceptance rate the speed profiles distribution generated is almost identical with the ground truth one having a Bhattacharyya distance of 0.0346. Such accuracy is appropriate for the purpose of estimating speed profiles for micro-pedestrian simulations.

CHAPTER THREE

3. People Counting

3.1 Introduction

In the previous chapter, a framework for measuring the speed profiles of pedestrians walking unimpeded was presented. However, in order for a simulation framework to achieve satisfactory precision in matching the real environment, in addition to the speed profile distributions, the knowledge of the amount of people entering and exiting the environment, and of the footfall of pedestrians at specific locations is necessary. Moreover counting people can provide useful information for monitoring purposes in public areas, assist urban planners in designing more efficient environments, provide cues for situations that might endanger the safety of civilians, and also be used by shopping mall and retail store managers for evaluating their business practices. Such knowledge can be obtained by analysing image and video footage from location specific cameras with goal to measure the number of people in them. For this reason in this chapter, a method for counting people in images and video sequences, from a fixed camera is presented.

While for measuring the speed of pedestrians, as presented in chapter 2, the localisation of the heel of a person, both in spatial and temporal dimensionality, is necessary; counting is better achieved by localising spatially the heads of the pedestrians present in an image and then sum the head detections to measure the total count. It is not coincidence that an expression such as 'headcount' and the idiom 'count heads' exist. Furthermore since our interest is in measuring the count of people using fixed cameras, the spatial knowledge of the background characteristics, since these we assume that remain unchanged, can assist us to distinguish better the presence or not of humans in an image.

The main contribution of this chapter is the development of a convolutional neural network (CNN) [80] that uses global image information for people counting and the use of temporal information for enhancing the precision in the obtained results.

In Section 3.2 a background study on the methods for people counting is presented, while in Section 3.3 the methodology of this approach is described. Finally in Section 3.4 the results and a critical discussion of our methodology is given followed by the concluding Section in 3.5.

3.2 Previous Work

Counting methods can be mainly categorised in to two groups. Counting by detection and counting by regression. In the former case, shape models of humans are tried to be fitted in different parts of the image to infer the number of people present in it, while in the latter, the number of people present in an image is being measured by learning the relationship between a distribution of low level features in the whole image and the number of people in it. It is obvious then that these two methods can be combined, as a person detector can be used to create a footprint on a distribution describing the whole image, which then can be used to infer the number of people in it, and as such hybrid methods do also exist. The use of CNN for the task of people counting is by its nature such an approach.

In the following subsections a discussion and presentation of the state of the art for each of these categories is being given.

3.2.1 Counting by detection

In counting by detection [42][89][103][118][127] the idea is to detect the presence of people in an image and then sum the detections to produce the final count. Detections of people is achieved with the use of object detectors such as histogram of oriented gradients (HOG) [33], poselets [18], edgelets [137] and others which describe a shape model of a human body using pixel information.

Such an approach is presented in [127] where a generic head detector is being used to estimate the number of people in an image. Initially gradient information is computed on a grayscale image. Based on the values of the gradient magnitude and orientation of a pixel, a binary image is created and by applying component labelling the centroid of each component is identified as point of interest. In order to further reduce the candidate

locations for the spatial position of the human heads present in the image, two background subtraction techniques [5][139] are being used. Having estimated the interest points, features based on gradient information and LUV channels in a region around a point of interest are computed using integral images [39]. Finally, by providing positive and negative samples, Adaboost is being used to create a soft cascade of classifiers to detect the presence of a head. A drawback of such approach is that it fails to deal with occlusions, thus it is appropriate only for camera views where the heads of pedestrians are fully visible. The use of foreground segmentation will also make the detection of static pedestrians impossible, although they mention that this step is not compulsory and it is being used to limit the search space. However, skipping the foreground segmentation will increase the false positive detections especially in environments with cluttered background information.

Another approach of counting by detection is proposed in [103]. Initially a HOG detector is used to create a probability distribution over the image with higher probability denoting the presence of a human. In order to deal with occlusions, the HOG detector is trained to learn only the upper part of the human body. Next the optical flow between two consecutive frames is computed. Assuming that the upper human body exhibits a uniform motion in contrast with the motion generated from the limbs, a mask resembling the shape of upper human body, is scanned through the optical flow response and a probability distribution of uniform motion is computed. The probability distributions learned from the shape model (i.e. HOG responses) and the uniform motion model are then combined and the fused probability distribution is searched, using Mean-Shift Mode Estimation [29], to localise head detections. Furthermore in order to increase the robustness of their system the mean motion vector is being used to establish valid human trajectories, and recalculate the detected pedestrians. Finally in order to deal with false detections generated from shape textures resembling to heads (e.g. a bag carried by a person) objects with coherent motion are identified, and the object with maximum height is kept, while the others are discarded.

Counting by detection is far from trivial problem, especially for objects classes like humans where there is big intra-class variation due to the non-rigidity of the human body. Moreover in visual surveillance applications where the camera may not have a top down view, overlap of objects and occlusions makes the task even harder. However human

detectors that can detect parts of human body have been proposed in the literature [18][42] and using also learned trajectories [89][118][137] can assist to deal with these. A pitfall of using counting by detection techniques is that they do not perform well in images with low resolution, since objects, in these, appear small and they do not generate enough information in order to be detected. Moreover, since most of these approaches use a sliding window to scan the whole image multiple times in different scales, they are computational heavy and thus slow.

3.2.2 Counting by Regression

In counting by regression [1][24][25][30][43][57][75][83][113][116], a mapping from some low level image characteristics, like edges or corners, to the number of objects is learned using some machine learning methods. Although using this approach the hard task of detecting an object is eliminated, ambiguities are presented as objects of other classes being present in the image might generate responses that will affect the counting. Furthermore since some of these approaches do not take into consideration the location of the objects in the images, the training phase requires a large amount of data, in order to cover all the possible perspective nonlinearities of the image plane. However annotating the ground truth data is a simple process since just a number with the objects present is required for each image.

In [116] a counting by regression method is presented, in which, the relationship of the features extracted from a part of an image with the number of people in the same part is learned. Initially an adaptive optical flow technique [35] is being applied and the foreground is segmented. Using the foreground as a mask, the number of edge responses, as well as, an edge histogram for each foreground blob is computed from the image. These along with the blob's spatial attributes are then being used as the feature descriptor of the blob. To deal with the perspective distortion effect the technique used in [24] is being applied. Blobs are annotated with the count of pedestrians they contain. In the case which a person is fragmented into multiple blobs, the contribution of that person to the total people count is split across the blobs that contain parts of him in direct proportion to the number of pixels contained in each blob. Finally a neural network is being trained with input nodes the features describing a blob and output the people count for the same blob. The total

count of all people in an image is thus the summation of the neural networks responses of all the blobs in the image.

In [83] a general framework for counting objects in images and videos is presented. The main idea of this approach is that integrals of density functions over pixel grids should match the object counts in an image. It is assumed that each pixel is characterised by a discretized feature vector, as in [116], and the training data are dot annotated. In the case of counting pedestrians in a video sequence, the pedestrians were annotated by a dot on their torso. Each annotated pixel then is being characterised, using a randomised tree approach [94], by a feature descriptor combining the modalities of the actual image, the difference image and the foreground image. For each pixel, a linear transformation of its feature descriptor is learned, using a random forest [20] to match the ground truth density function, which is calculated as summation of isotropic Gaussian kernels centred at the dot annotations of the objects present in the image. The learning of the random forest depth is achieved using the Maximum Excess over SubArrays (MESA) distance loss metric.

Inspired by [83], in [43] a similar approach is presented, where instead of using a feature descriptor per pixel, patches of dense features of ordinary filter banks are computed, and a regression forest is used to learn the mapping of these patches to the desired density function.

The approach in [24] is consistent to the objective of protecting people's privacy. A video sequence is represented as a collection of spatio-temporal patches from which initially a mixture of dynamic texture model [23] is learned using the Expectation Maximisation (EM) algorithm[93]. The learnt model segments the foreground information and also discriminates between crowds moving in different directions. In order to deal with the perspective distortion problem, each pixel is weighted using linear interpolation based on the distance from the camera. Thus pixels capturing information of the environment which is closer to the camera, i.e. objects appear large, are given less weight than the once that that represent information of more distant parts of the setting. The segmented part of the video is then characterised using spatial features capturing the segmented area information, edge and texture features. Finally a Gaussian Process (GP)[113] is used to regress the feature vectors to the number of people per segment. For each class segmented a different process is learned, using a combination of linear and Radial Basis Function kernels.

In [75] a mixture of Gaussians is initially applied to extract foreground information. Then using the foreground as a mask on the original image, edges are detected using the canny edge detector. Histograms of the area of the foreground blobs and edge orientation are then being used as features to describe the image. By using the homography between the ground and the image plane a variable sized cylindrical model, representing a pedestrian, is being used to normalise the features. Finally by using a feed forward back propagation neural network with inputs the histograms of the normalised features, a relationship between the features and the number of pedestrians in the image is learned.

In [25] a counting by regression technique based on local extracted features is presented. They argue that accurate crowd counting is based on localised feature importance mining and visual information sharing among spatially localised regions. In the first step the perspective normalisation technique used in [24] is employed to create a weight map for each pixel. Then each frame is partitioned in a number of cells, and for each cell a feature descriptor is computed using segment-based, structural-based and local texture features. A frame is then described as the concatenation of these features of all cells in that frame. Moreover by annotating each cell with the number of persons present in it, a multidimensional output vector is created for each frame, constituted as the concatenation of the output responses of all cells in a frame. Thus the problem of counting the number of people in an image is becoming a regression problem, having to learn the relationship between the feature vectors and the output responses. In order to deal with the colinearity that some low level features might exhibit, thus leading to the overfit of the parameters of the model, the multivariate ridge regression function [121][54] is being exploited. Their model allows the exchange of information between cells and capturing the importance of a local feature, by taking into consideration all the cells of a frame while computing the weighted importance of single cell. Counting of pedestrians then in a single frame is the sum of the outputs of the learned regression functions from each cell. Since this method doesn't employ any object detector, it assumes that all the foreground objects are humans and thus might produce inaccurate measurements in the presence of other object classes or when pedestrians are presenting some abnormalities (i.e. humans with baggage, trolleys, etc.). Moreover it doesn't exploit the richness of the spatio-temporal information and thus occlusions of pedestrians cannot be detected leading to false output from the regression model.

In [1] a method for counting people inspired by the Harris corner detector [56] is presented. Initially corners in an image are detected by using the eigenvalues of the covariance matrix of gradient information for each pixel. Using a multiresolution block-matching technique [131], they compute for each corner a motion vector, and thus they differentiate between static corners (i.e. null motion vector) and moving ones. Assuming that each person in the image generates the same amount of moving corners, the number of people in a frame is therefore computed as the result of the division between the moving corners detected and the average number of corners per person. Taking into account only the moving corners, this approach fails to recognise static people. Also the camera perspective effect is not taken into consideration, thus in scenarios where the size of pedestrians varies greatly depending on their location, the assumption that each pedestrian generates the same amount of corners responses is invalid, producing erroneous estimation in the total count of people.

Inspired by [1], the approach in [30] uses low level features to estimate the number of people in a frame. Initially SURF interest points [7] are computed for each frame. Then, the detected points are partitioned into clusters using a graph based clustering algorithm [44]. The distance of a person from a camera is estimated assuming that the lower points of a cluster lie on the ground plane. Moreover people are assumed of being of average height and in order to learn the association between the numbers of SURF points detected and the position of a person in the image plane (i.e. perspective effect), inverse perspective mapping is applied. To deal with occlusions the density of each cluster is estimated, since small distances between interest points and large cluster areas indicates the presence of more than one person in a cluster. Each cluster is then represented by a feature vector, comprising of the number of SURF points present in the cluster, the distance of the cluster from the camera and the density of the cluster. Using a variation of the Support Vector Machine a relationship is then learned between the feature vector of a cluster and the number of people present in it, and the initial estimate is averaged over a moving window for a number of frames.

A drawback of all regression approaches is that they cannot discriminate well between intra-class variations (i.e. differences in human sizes, humans carrying objects, humans with bicycles etc) and since they lack learning shape models, they are unable to

differentiate between interclass (e.g. animals) differences. Thus their application is mostly location specific as an environment with large variance in content will cause them produce erroneous measurements.

3.2.3 Hybrid Approaches

Hybrid methods that combine counting by detection and counting by regression [62][108][31][144] attempt to eliminate the drawbacks of both approaches by fusing their techniques.

For example, the technique of [83] is used in [108] where a hybrid of counting by detection and regression approach is presented. Each pixel is represented by a binary feature vector. Using an object detector [42] a density image is computed where each pixel value defines the confidence output of the detector. This value is then discretized and represented by a binary feature vector having the value '1' in the dimension of the corresponding discretized value and '0' in all the others. Furthermore SIFT features [85] are extracted from the image in order to compute another binary feature vector. A number of SIFT prototypes, equal to the number of dimensions of the feature descriptor computed using the detection method, are used to describe a pixel. The prototype that is the closest of describing the pixel has its value '1' in the feature descriptor while the others have the value '0'. The concatenation of the two binary feature vectors is then used to describe each pixel, and by minimizing the regularised MESA distance, the weight of each discretized feature is learned. The density of each pixel is thus calculated by multiplying its feature descriptor with the learned weight vector, and the count of people in the image is then estimated by the integral of the density of the image.

Another example of a hybrid approach is presented in [62], where a method to count and detect humans in video sequences in crowded environments is presented. In order to count humans, a Gaussian mixture model is initially applied on a grayscale video sequence to obtain the foreground information. Then in order to resolve the perspective distortion effect, and assuming that the size of an object varies linearly as a function of the y-coordinate of the image, the number of the extracted foreground pixels for each row of the image is being recalculated. The foreground information obtained is further processed using a closing operation. The size of the kernel applied varies in size based on the y-

coordinate of the pixel on the image plane. Counting of humans is then becoming a problem of finding a relationship between the number of foreground pixels and the number of humans present in the image, a relationship which is learned using a neural network. To detect individuals, KLT features are initially extracted from the image using as mask the foreground information. A combination of an ellipse descriptor and a Gaussian distribution is being used as a cluster model, again varying in size based on the y-coordinate of the image plane. Finally an EM algorithm, initialised using the information from the counting step, is being used to cluster the KLT features into the cluster models and thus localise the pedestrians. In this approach it is assumed that all people, even stationary ones generate some form of movement which is not valid. Also this approach is heavily dependent on the foreground information, therefore a slow adapting background model will cause false detections in the case of static people moving, and a fast adapting one will mistakenly categorise static pedestrians as background.

Finally the following two hybrid approaches [31][144] are the only ones, to the best of our knowledge, that use CNN for people counting. Both attempt to exploit the CNN characteristic of the spatial invariance in the detection of patterns, and thus the networks described are trained as human detectors by using spatial crops from whole images for training. In [31] a CNN is learned to estimate the density of people in an image by using crops from the full resolution training dataset. The learned network is then applied to the whole image information to produce a density map of human presence and moreover its parameters are transferred to two similar networks that are applied on different resolutions of the global image. The response from the three networks is then averaged to produce a final density map. To count the number of people in the density image, each point of the density estimated is fed to a linear regression node. The weights then of the node are learned independently for the density estimation. In [144] cropped images are also used for training, however the learning of the density and the total count is not serial, but takes place in parallel. Both the density map and the linear regression node are connected to the same CNN and learning takes place by altering the cost function between the one used for the density estimation and the one used for count estimation.

In the following section our methodology for counting people is presented. As mentioned, in the introduction of this chapter, the main contribution of this work, which distinguishes

it from the two approaches aforementioned, is the use of the whole image information for the creation of a spatially-variant model and the use of temporal information for further enhancement of the counting precision.

3.3 Methodology

Our approach following the methodology proposed in [31] tries first to generate a density map, indicating the presence of humans in the image under investigation, and then to learn the relationship between the distribution of activations and the actual count number by using regression. Furthermore a sequence of responses from time-lapsed images is fused to further increase the precision of the count estimate. Since a picture is worth one thousand words, the architecture of the whole network presented in this chapter, is displayed in Figure 3.1.

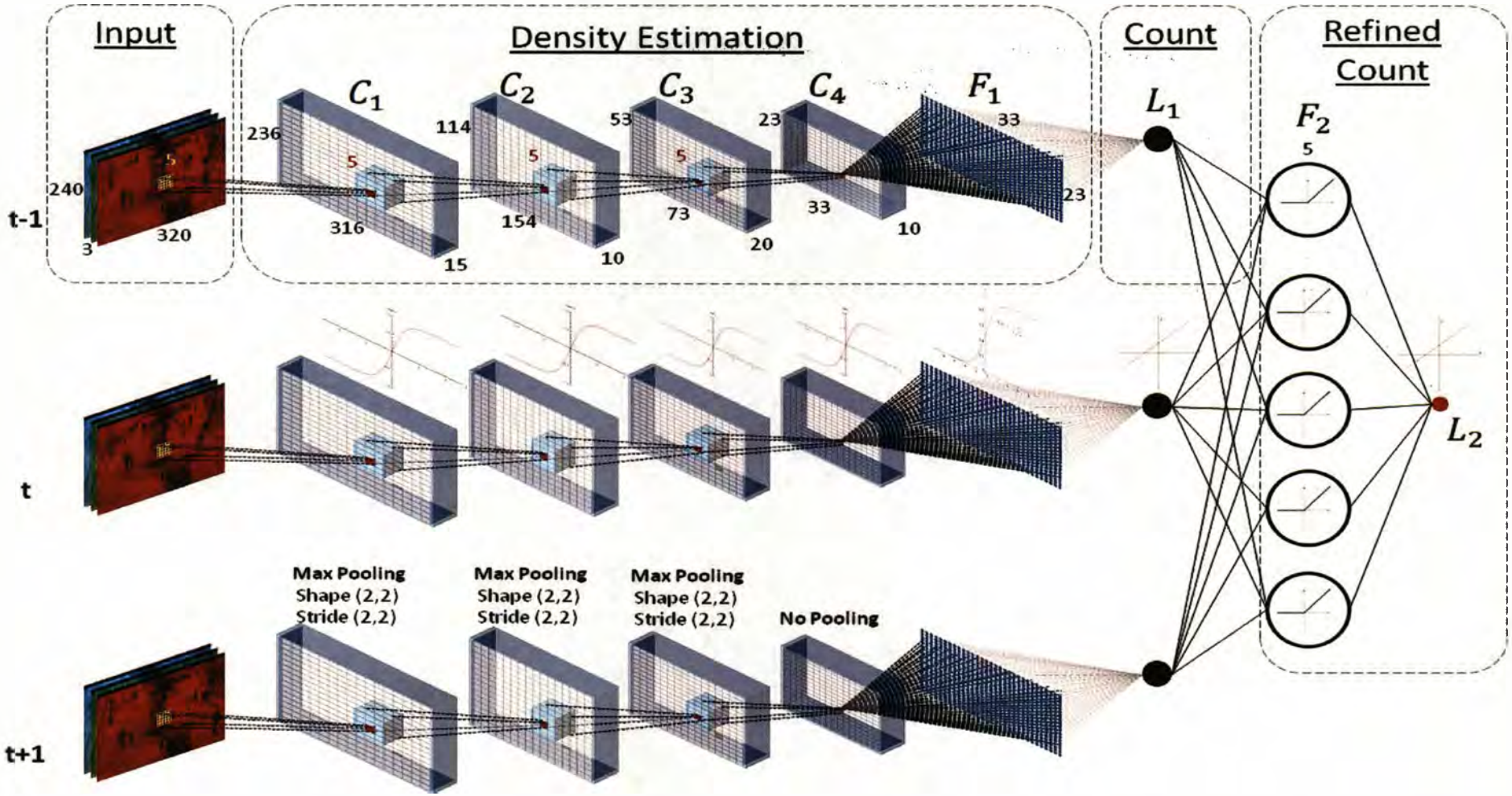


Figure 3.1: The proposed architecture for pedestrian counting. In the left we can see the temporal data input in a form of consecutive in time RGB frames, while for the density estimation a pipeline with 4 convolutional layers followed by a full connected sigmoid layer having the task to produce the density images. For the count of a single pipeline a linear regression unit combines the 759 inputs to produce a final result. Finally by combining the results from the counts of 3 pipelines in full connected rectifier layer we feed a node to perform linear regression and produce the final result.

From the previous figure, we can see that there are three main pipelines and four processing steps in the proposed network. The pipelines share identical parameters values and thus only one is needed to be trained in order to reproduce the others. The first step which is the input to a single pipeline will be discussed in Section 3.3.2 while in section 3.3.3 the architecture of the CNN will be described. In Section 3.3.4 the part of the pipeline for the count estimate will be discussed while in Section 3.3.5 the fusion of the result from the three individual pipelines will be presented. For the ease of the reader in Section 3.3.1 the structure of a convolutional layer is introduced.

3.3.1 Convolutional Layers Primer

The functionality of a convolutional layer can generally be described as the application of a set of filters/kernels on an input signal, to produce a series of outputs (features) where each one of those describes the localised response of one filter on the whole signal.

In the case where the signal is a digital image, the input can be seen as a 3D rectangular volume with size (w,h,c) where w is the width, h is the height and c the number of channels in that image. As such, the kernels applied on an image occupy as well a 3D rectangular volume of size (d,d,c), thus a kernel covers the same 2D spatial location at each channel in the image. Let's for simplicity assume that the input is a grayscale image with one channel, and that the size of the kernel to be applied is 5x5. For each kernel application on the input image a response is recorded to a neuron of a convolutional feature (Figure 3.2) and by sliding the kernel on the whole image a map of responses is generated.

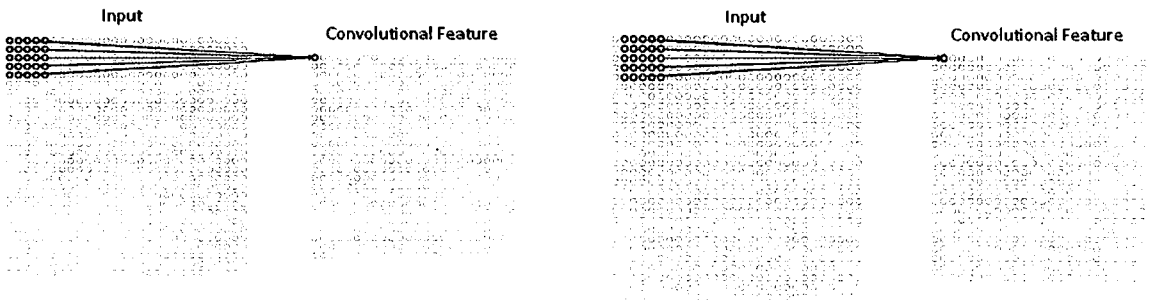


Figure 3.2: Receptive fields of two adjacent neurons in a convolutional feature (image from [96]).

As it can be seen in Figure 3.2 neighbouring neurons in a convolutional feature describe neighbouring spatial locations on the input image, and as such the value of the neuron at j column and k row in a convolutional feature is equal with

$$n_{j,k} = f(b_{j,k} + \sum_{l=0}^{4}\sum_{m=0}^{4} v_{l,m} \cdot a_{j+l,k+m}) \quad (3.1)$$

Where f is an activation function that takes as input the bias $b_{j,k}$ associated with the j,k neuron, $v_{l,m}$ is the value of the filter in it's (l,m) location and $a_{j+l,k+m}$ denotes the input value at the location $(j+l,k+m)$. In the case where the image has more than one channel then another summation over the channels should be added in Equation 3.1.

A convolutional layer can have multiple features, with each one being characterised by its own filter and the output of a convolutional layer, as seen in Figure 3.1, can become the input of another convolutional layer, thus in general the produced volume of a convolutional layer has a size of width W_o, height H_o and depth D_o which are estimated given by the following equations

$$W_o = \frac{W_i - S_k + 2 \cdot P}{L} + 1 \quad (3.2)$$

$$H_o = \frac{H_i - S_k + 2 \cdot P}{L_k} + 1 \quad (3.3)$$

$$D_o = N_f \quad (3.4)$$

Where W_i and H_i is the width and height of the input volume to the convolutional layer respectively, S_k is the spatial extent of the kernel, P is the amount of zero padding, L_k is the stride of the kernel and N_f is the number of different kernels, and thus features, used in the layer.

It is common after calculating a convolutional feature to subsample it by performing a pooling operation, before feeding as an output to subsequent layers. This is performed in order to reduce the number of parameters and thus the computation of the network. Pooling operates independently on each feature in the convolutional layer. Most commonly a pooling layer with filters of size 2x2 with a stride of 2 is applied on each feature selecting only the maximum value thus discarding 75% of the feature activations. Assuming that a pooling operation, with a pooling filter of spatial extent S_p and stride L_p, is performed after the

produced volume estimated from Equations 3.2-3.4 then the output volume of the pooling operation will be

$$W_p = \frac{W_o - S_p}{L_p} + 1 \tag{3.5}$$

$$H_p = \frac{H_o - S_p}{L_p} + 1 \tag{3.6}$$

$$D_p = D_o \tag{3.7}$$

For more information on the convolutional layers and their structure the reader is referred to [96].

3.3.2 Input

The performance of a supervised neural network is dependent mainly on following three factors: a) the input data that is presented, b) the network's architecture and its parameters, and finally c) the ground truth data which represent the task that the network is asked to learn. As such appropriate representation of the input can lead to better and faster learning of the network [96]. The input layer of a single pipeline, as seen in Figure 3.1, is an RGB image of size 240x320 pixels. Every frame, is pre-processed by initially calculating the mean in all training images and subtracting it from all the pixels, before entering the network. Then data is centred around zero in all dimensions and scaling in values between -1 and 1 is performed applying Equation 3.1 on each pixel:,

$$p_{x,y,c,s} = 2 \cdot \frac{p_{x,y,c,m} - \min(f_t)}{\max(f_t) - min(f_t)} - 1 \tag{3.8}$$

where $p_{x,y,c,m}$ is the pixel value of frame f_t at location x,y of channel c, after the mean subtraction and $p_{x,y,c,s}$ is the pixel value after the scaling which we will refer from this moment as $p_{x,y,c}$. The reason that data is zero centred is to facilitate learning of the network [81] and specifically for the gradient descent algorithm to avoid zigzagging while minimising the cost of the network. Furthermore data is desired to be scaled having thus obtained relatively small values in order to balance out the rate which the weights connected to the input node learn.

3.3.3 Density Estimation

The density learning pipeline, as presented in Figure 3.1, is comprised of four convolutional layers followed by a fully connected one. More specifically for the convolutional part of the

density estimation pipeline, C_1 has 15 features of size 316x236, C_2 has 10 features of size 154x114, C_3 is comprised of 20 features of size 73x53 and finally C_4 has 10 features of size 33x23. The detection kernel of all convolutional layers is 5x5 with a stride of 1 and the feature activations, except from those of C_4, are max pooled with a kernel of shape 2x2 and stride of 2; thus halving each dimensionality of a feature before feeding it as an input to a subsequent convolutional layer. [31], where all activations in a feature share the same bias, each feature activation in our implementation is characterised from its own bias. Since our input is the whole image we want to allow our network to further tune the importance of a feature to a spatial location. Following the notation of Equation 3.1 the activation function applied for a neuron belonging to a feature f in our CNN is the hyperbolic tangent given by

$$a_{f,j,k} = \frac{1 - e^{-2 \cdot z}}{1 + e^{-2 \cdot z}} \tag{3.9}$$

$$where \;\; z = b_{f,j,k} + \sum_{f_{-1}} \sum_{l=0}^{4} \sum_{m=0}^{4} v_{f,l,m} \cdot a_{f-1,j+l,k+m} \tag{3.10}$$

where the leftmost summation in Equation 3.10 sums over all the features present in the previous layer (as mentioned before, in the case where the previous layer is the input image, each channel of the image represents one feature).

The last layer of the density estimation pipeline is a fully connected one (F_1 in Figure 3.1) which has as many neurons as there are present in one feature of the previous layer (i.e. C_4). Each neuron in F_1 is connected to all the neurons present in C_4 and thus the weight vector v_i of each neuron i has 7590 (33x23x10) dimensions. The activation function used for each neuron of this layer is the sigmoid thus the following equations apply.

$$a_i = \frac{1}{1 + e^z} \tag{3.11}$$

$$z = b_i + \sum_{f=1}^{10} \sum_{l=1}^{33} \sum_{m=1}^{23} v_{i,r} \cdot a_{f,l,m} \;\; , \qquad r = 759 \cdot (f - 1) + 23 \cdot (l - 1) + m \tag{3.12}$$

F_1 is the last layer in our density estimation pipeline and as such the 33x23 responses a_i of the layer are compared against the equivalent y_i of a ground truth density of same dimensionality in order to measure the error which will be back propagated for the learning.

As such the cost function we are using for the comparison is the Kullback–Leibler divergence shown in Equation 3.13, and the error produced is the mean cost across all the examples seen.

$$KL(y_i||a_i) = y_i \cdot log\frac{y_i}{a_i} + (1 - y_i) \cdot log\frac{1 - y_i}{1 - a_i} \tag{3.13}$$

3.3.4 Counting

After our network has learned to estimate a density denoting the human presence, we are interested in finding the relationship between this density and the actual count of people. For this reason a single linear neuron (layer L_1 in Figure 3.1) is fully connected with the sigmoid neurons of F_1 .Learning is performed by linear regression using the mean squared error across a number of examples as cost function. Thus if a_i denotes the *i*'th activation from layer $F_1$and v_i the entry in the weight vector of L_1 associated with a_i, and b_c the bias and a_c activation value of L_1 then,

$$a_c = b_c + \sum_i v_i \cdot a_i \tag{3.14}$$

and the cost for a single example, when y is the ground truth count, is given by $(a_c - y)^2$.

3.3.5 Refined Counting

The accuracy of people counting, can be further improved by fusing measurements from networks operating on subsequent frames along the temporal dimension. For this reason, three pipelines operating on frames with timestamps *t-1*, *t* and *t+1* are fully connected to a vector of five rectified linear units. Each rectified neuron has as activation function similar to Equation 3.14 with the only difference that negative values, produced by the summation of the weighted input with the bias, are producing an output of zero.

Finally all five outputs from the rectified linear units are connected to the linear neuron L_2 for the refined count. The only difference in the linear regression performed in this neuron compared to the one in L_1 is the cost function, since for this we are using the absolute difference of the estimated count against the ground truth.

3.4 Results

In the previous section, the basic elements, as well as the whole structure, of our network was described. The network was implemented using Python and more specifically the pylearn2 [49] and theano [6][11] machine learning libraries which provide automatic gradient descent functionality and the building blocks to create the network. This section describes the relevant experiments using in detail. More specifically in Section 3.4.1 information regarding the dataset used is provided while in Section 3.4.2 a presentation of two comparative methods is given while in Section 3.4.3 the experimental configuration is discussed , followed by Section 3.4.4 in which the results are presented.

3.4.1 Dataset

For our experiments we used the publicly available Mall crowd counting dataset [86][111], a representative frame of which can be seen in Figure 3.3. The particular dataset was selected as the pedestrian simulations could be applied in a Mall environment, as it will be further discussed in the next chapter.

Figure 3.3: A frame from the Mall dataset [86]

The dataset consists of 2000 time-consecutive frames of 640x480 resolution with the frame rate around 2Hz. Over 60.000 pedestrians are annotated, with a point indicating their head location, and the whole dataset is being recorded from a fixed camera in an indoor shopping mall environment. It is a challenging dataset with constant movement in the space, where pedestrians wander freely, alone or in groups, forming a cluttered environment with plenty of occlusions. Moreover reflections are generated in both the windows of the shops and the floor, the lighting conditions change, and the viewing angle of the camera causes pedestrians to vary in scale along the row dimension of the images.

The dataset except from the ground truth annotations provides a perspective map where it denotes the relative scale of pixels in the three dimensional scene (Figure 3.4)

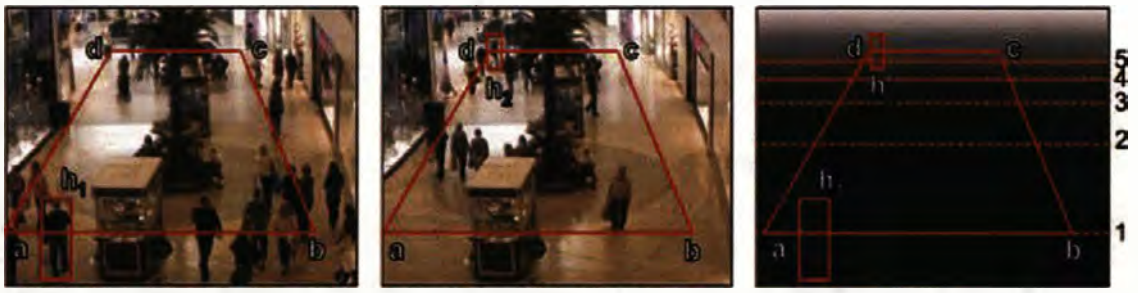


Figure 3.4: By measuring the size of people in different time frames (a), (b), the perspective map denoting the relative scale of pixels in the real word dimension.

3.4.2 Competitive Methods

We also implemented the only two other methods [31] [144] in our knowledge that perform people counting with the use of Convolution Neural Networks to enable comparison to our method.

In [31] a CNN is created to perform people counting. Their network architecture can be seen in Figure 3.5.

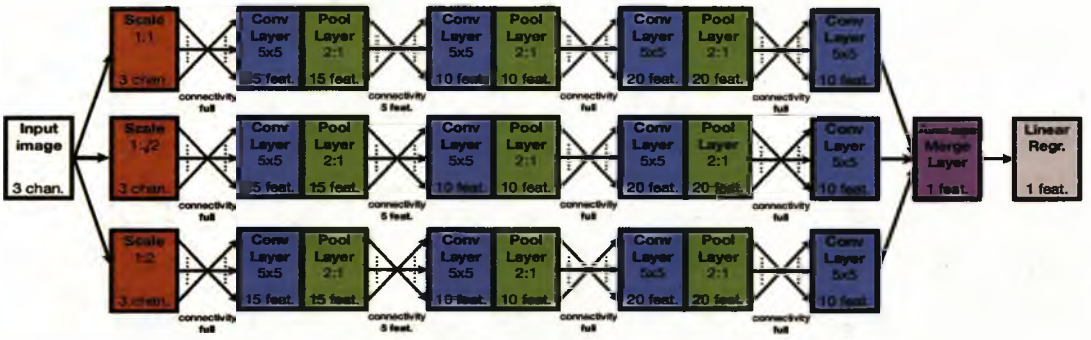


Figure 3.5: Architecture of network for people counting presented in [31]

Similar to our approach, the three main pipelines of their architecture are identical in their configuration and in their parameter settings. However they apply each pipeline at different scales of the images in order to infuse scale invariance in their network. To train a single pipeline they use cropped images from their training dataset thus trying to learn a location invariant person detector, which then can be applied to the whole image for density estimation. The ground truth they use for the density estimation is binary images with the foreground information denoting the head location of a person, while for the regression counts of people in the training images is used. Since the scale of the input images in the pipelines is different, and thus the size of the convolutional features too, they use one bias per feature in contrast to our approach where every node in a feature is associated with a unique bias.

Each pipeline estimates a density, representing the human presence in the environment, and the average merge layer in their network merges the three different density estimations into one followed by a linear regression node for the count estimation. To merge the three densities, derived at different scales, they use the following equations.

$$y(i,j) = \frac{1}{1+e^{z}} \tag{3.15}$$

$$z = \sum_{s\in S}\sum_{n\in F}\frac{1}{S\cdot F}x_n^{(s)}(i',j') \quad , \qquad i' = \frac{i}{R_{s,i}}, \qquad j' = \frac{j}{R_{s,j}} \tag{3.16}$$

where $R_{s,i}, R_{s,j}$ are the ratios between the feature map dimensions at the highest scale and at the s-th scale, S is the number of scales and F is the number of feature maps. Thus each node

$x_n^{(s)}$ in a single density has equal importance to the construction of the merged one. The activation function they use throughout a single pipeline is the one from Equation 3.9.

The second method we implemented is the one presented in [144] and its architecture can be seen in Figure 3.6

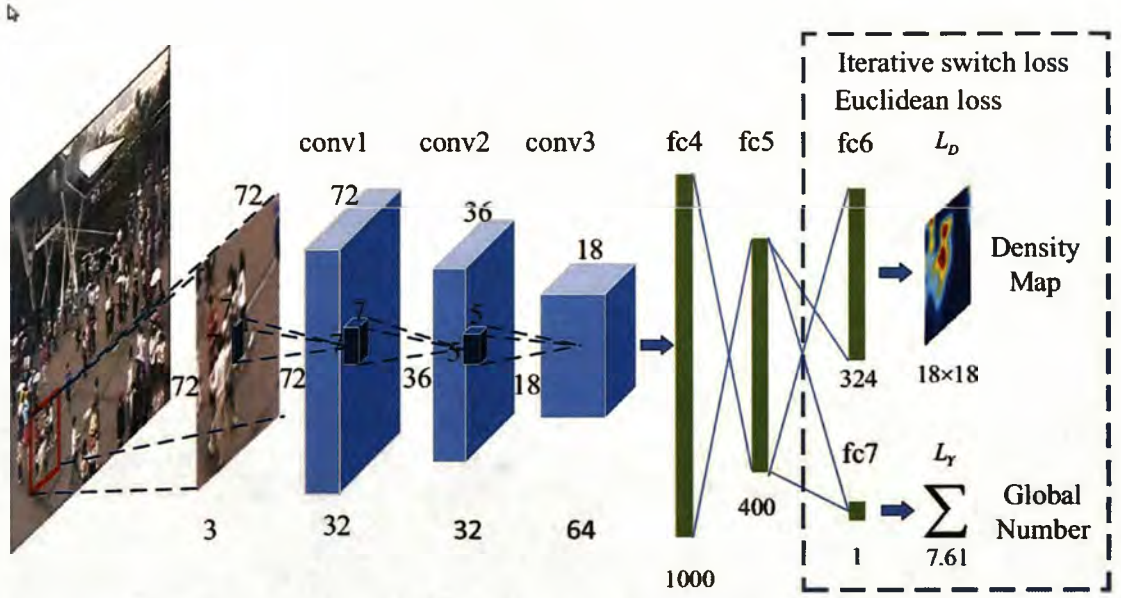


Figure 3.6: Network architecture for people counting presented in [144]

In this approach similar to the one in [31] cropped images are being used for the training of their network, however the network learned is being applied on the whole image in a sliding window fashion (instead of being applied in the whole image as in [31]), where each detection window generates a local density, and thus the density estimate for the whole image is calculated by creating a mosaic from the use of the local ones. Throughout their network the activation function being used is the rectified linear, and one interesting point in their architecture is that instead of learning a density and then perform a linear regression to estimate the count, the training of the density and the counting takes place in an alternate way. Thus, while training the network for density estimation, and when a threshold value in the error difference between consecutive training iterations is reached, the network's cost function changes by replacing the density estimation layer with the count estimation layer. Then, again when the error is not improving while performing training for the count, the count layer is replaced with the density estimation. The layers are alternated until both cost

cannot be further improved, and in every switch the parameters of the network, prior to the count or density estimation layers, remain unchanged, thus transferring their knowledge in the new setup. For the density estimation learning the ground truth used is a density image created from the responses of a Gaussian distribution, centred at the head of a person, and a bivariate normal distribution, placed at the body of the person. The combined distributions describing a person are then normalised to add up to one. After performing the previous step, the ground truth for the counting is then just a summation of the entries in the ground truth density image.

3.4.3 Experimental Configuration

The ground truth we used for training our network was based on the annotation points of the dataset. By measuring in pixels the width of the pedestrians' heads in different locations we learn the association between image row and head size. Then centred at the annotated points, squares with pixel values of 1 represent the head of the pedestrians while all other locations have value of 0. Since the density estimation resolution in our pipeline is 33x23, the generated binary images of 640x480 were scaled down and each image was normalised to have values in the range between zero and one. The network was trained for 500 epochs. Using the same training termination criteria for the method of [31], the ground truth was based on cropped images of size 320x240 from the original 640x480 binary images created in the previous step, scaled to a resolution of 33x23 and normalised with values between 0 and 1. For the method in [144] the ground truth density images were generated by using a Gaussian kernel summing to one, centred at each annotation point and with a standard deviation based on the values of the perspective map of the dataset. Crops of size 72x72 from the 640x480 density images were then extracted and scaled down to size 18x18 by preserving though the total summation of the density to match the one before down sampling. For the training process of this approach we used 70 epochs for each iteration of a cost function. In Figure 3.7 samples from the input images along with their ground truth densities are displayed.

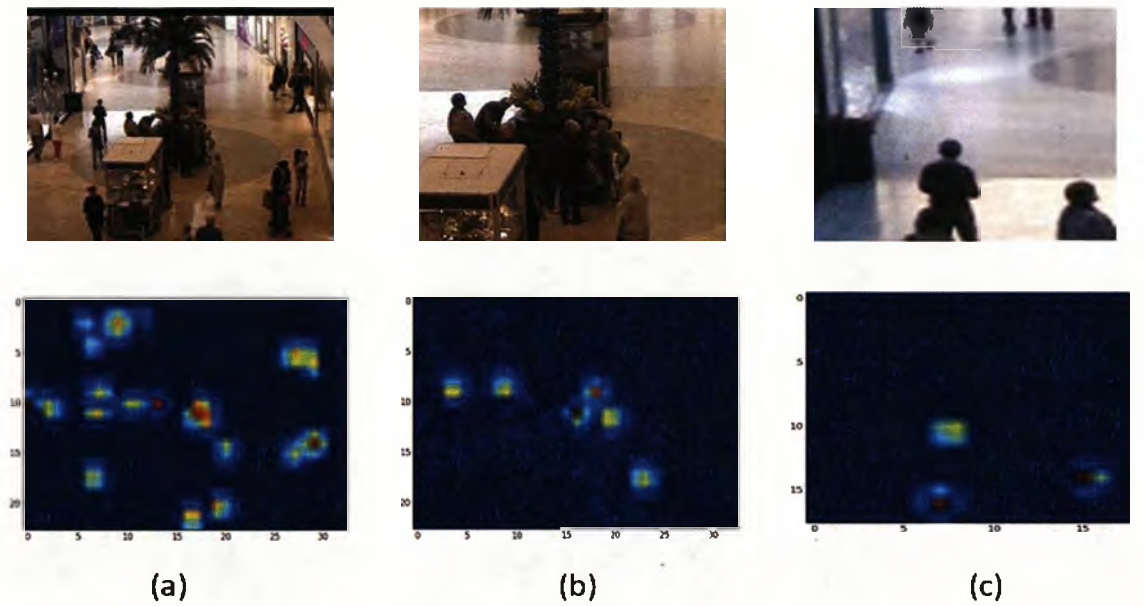


Figure 3.7: Input images for training to the three different networks and their associated ground truth from the same frame.a) Our approach using as input the whole image information (resolution of 320x240) and ground truth of size 33x23, b) the approach from [31], using as input a cropped image (size 320x240) from the whole frame (resolution of 640x480), and ground truth of size 33x23, c) the approach from [144], using as input a cropped image (size 72x72) from the full resolution whole image (resolution of 640x480) and ground truth of size 18x18.

From the 2000 frames of the dataset, 1000 were used for training 250 for validation and 750 for testing. For [31] we used 5 cropped images (size 320x240) per training whole image (640x480), while for [144] we extracted 50 cropped images. All cropped images are selected randomly. The input image resolution we used to test out methodology is 320x240.

Training a CNN means fine tuning various parameters. However some of the training parameters were kept constant through all the experiments. The dropout rate [125] was fixed to 0.5 for all layers. This means that during training each node has 50% chance to be activated, and its parameters to get updated, this assists for regularisation and thus avoiding overfitting the network parameters to the training dataset. Another parameter we kept constant was pooling, by always using the same pooling kernel with same stride. Also all weights were initialised using a uniform distribution and with range (-0.05, 0.05). Other parameters however, such as the learning rate, the use of momentum [130], the maximum norm of the weight vectors were selected separately for each experiment by testing their impact on the learning behaviour of a network on small subset of the training dataset. The algorithm used for the training was stochastic gradient descent with mini batches. Thus the

update of the network parameters occurred regularly, after seeing just a subset of the training dataset and not at the end of each epoch (i.e. estimating the cost after seeing all training data once).

3.4.4 Experimental Results

In Figure 3.8 we can observe the density estimation results from the different methodologies. As we can see, our approach manages to describe quite satisfactory the distribution of the pedestrians in the space. In contrast the responses from [31] are not descriptive at all, since it appears that although there is a change in the density estimation from frame to frame it follows a general pattern, and it seems like the network failed to learn the people's density. Also the density results derived by [144], although more descriptive regarding the presence and the position of the pedestrians in the space than the one of [31], still generates many false positive activations.

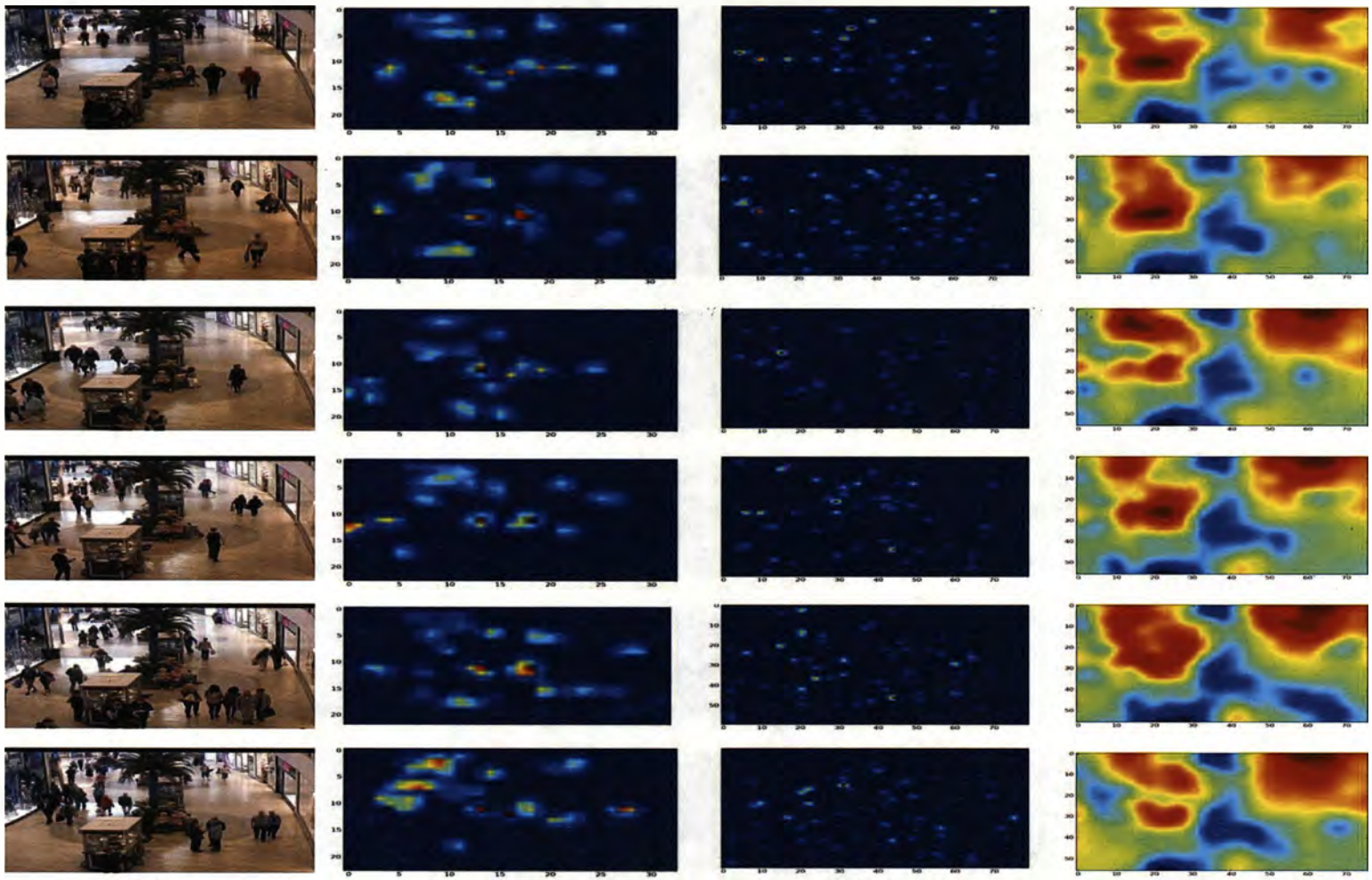


Figure 3.8: Density estimation results.(a) Input frame.(b) The response from our approach.(c) The response from [144] .(d) The response from [31]

Looking at the training and validation cost function outputs offers us a better insight. In Figure 3.9 the error of the training and validation dataset through each epoch of training for the approach in [31] is shown. First of all, it can be seen that for both sets the behaviour of the cost follows the same pattern and that almost after the first epoch the error has reached its minimum for both the training and the validation set. Afterwards, no further learning seems to occur and the error stays fixed at a certain value.

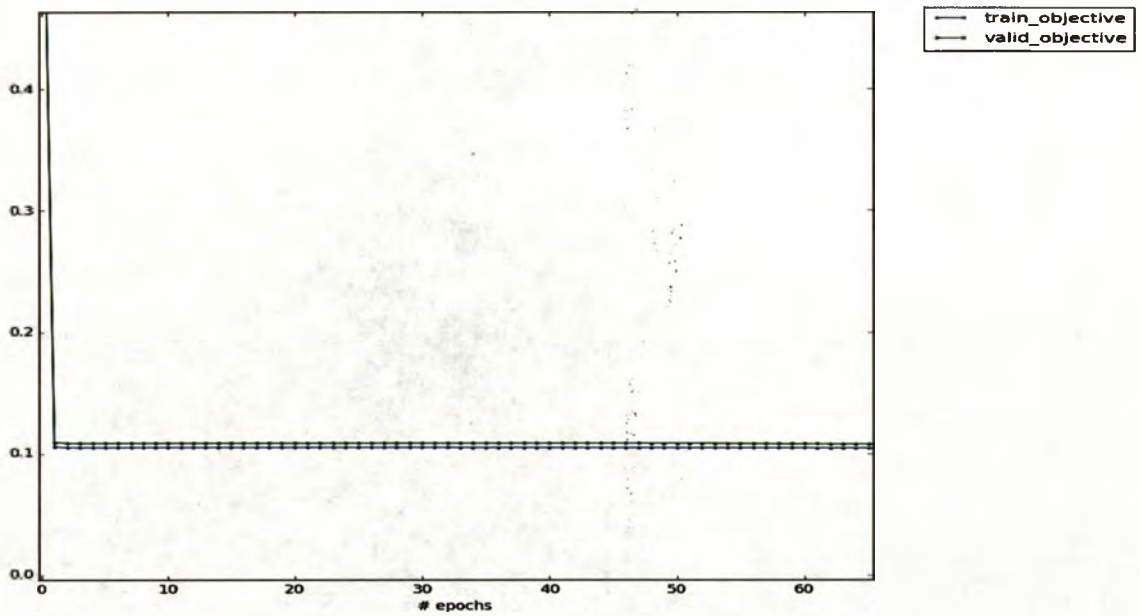


Figure 3.9: Cost function score for training and validation set per epoch for [31]. From the error graph we can see that the network learns just after one epoch and the validation and training error cannot be further improve. Although both errors are very close to each other, the network seems to learn a general solution which does not provide good accuracy.

Figure 3.10 shows the error for the training and validation set of [144]. In the graph we can observe that the error in the training dataset decreases again but it reaches its stalling point after the 20^{th} epoch. Moreover we can observe that the validation error decreases too through time but it is reduced in slower rate than the training error does. Although the network seems to learn, the knowledge that it gains is very specific and fails to generalise satisfactorily.

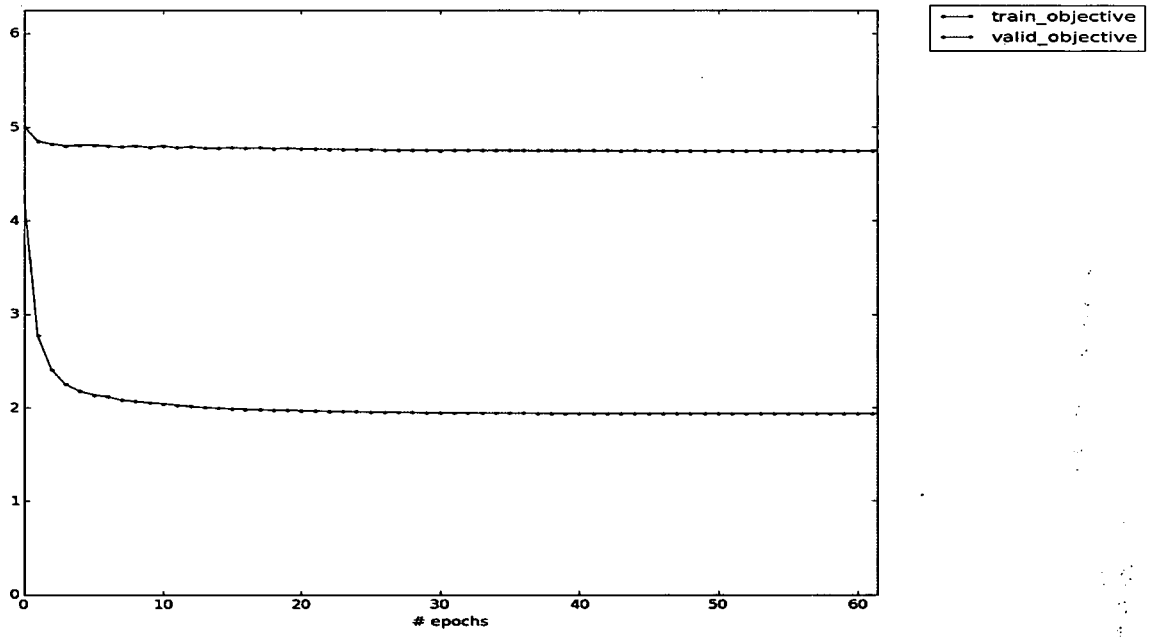


Figure 3.10: Cost function score for training and validation set per epoch for [144]. We can observe that both the validation and the training error decrease as epochs pass, however the network seems to not to generalize very well.

Finally Figure 3.11 shows the cost per epoch for our methodology.

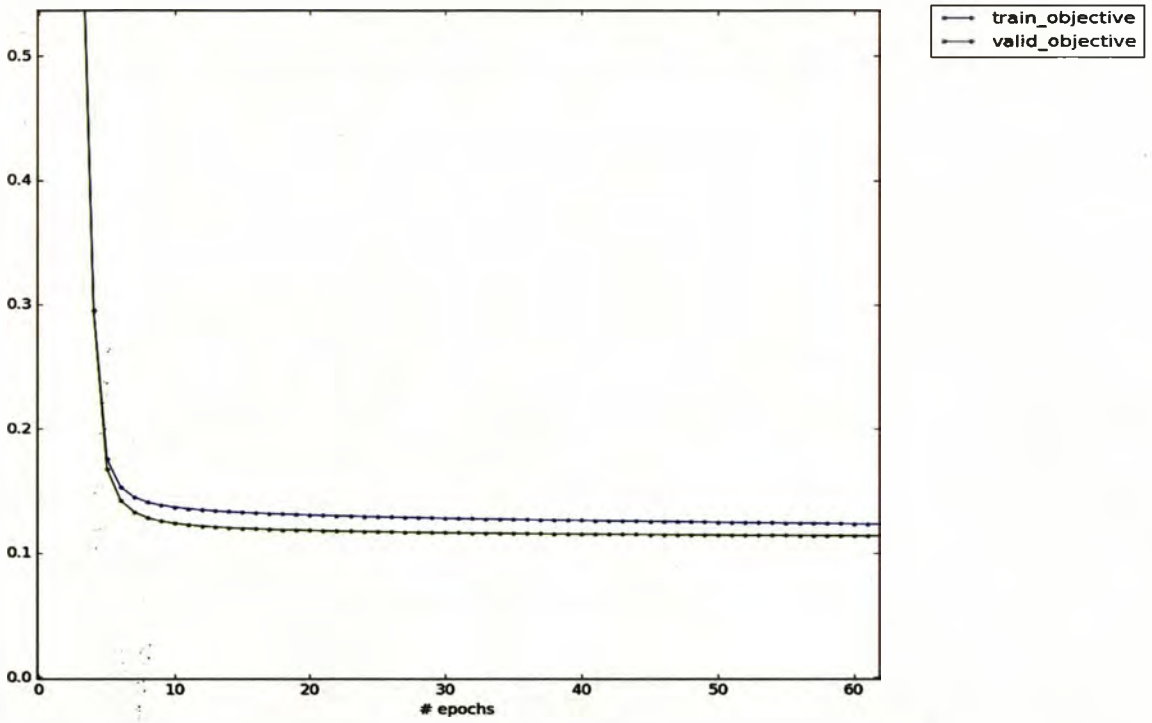


Figure 3.11: Cost function score for training and validation set per epoch for our methodology. Although our network doesn't learn as fast as the previous ones, it achieves to provide a generalize solution that it is accurate for the task of people counting.

From the above evidence, the error for both training and validation decreases at the same rate and after the 10^{th} epoch it declines slowly. Although the validation error seems to be lower than the training, as the training continues, the error in the training set eventually becomes smaller than the one for the validation.

Let's consider the number of parameters in the configuration of each network. The total free parameters for learning in the network of [31] is 14,930. While for the one in [144] the number of parameters that are available for learning is 21,373,532. In our proposed network the number of parameters is 5,871,954. Finally the difference between our approach and the other two is that we use for training input whole images while they use cropped images.

Based on the information provided above, our assumption is that the method of [31], not only has too few parameters to offer a solution to the problem, but also because it lacks any fully connected layer, no information is exchanged between the nodes that can result to a combination of features detected. On the other hand the approach in [144], has a plethora

of parameters to adjust and to solve the problem of detecting people in an image, and furthermore they exchange node information by using full connected layers. However by using cropped images as input it does not provide any spatial localised information that will facilitate learning the presence of the background in the whole image. Our proposed network, with almost a quarter of parameters compared to [144], assumes whole input images, combines the information from the various nodes of the detectors and therefore it can learn localised background/foreground information. In other words, if our task was to find a fly on a wall, the approach in [31] scans the wall to find the fly with a lens that makes things to appear very blurry and the presence of the fly is diffused on the wall, while the one in [144] scans the wall with a lens that can see every little detail, thus some irrelevant complex patterns of the wall may confuse it. In contrast to the other two methods, our approach avoids scanning the wall, instead it just removes it and observes what is left.

After the density is estimated, the next step is to perform the counting. The mean deviation error (MDE) ε of the counting step,

$$\varepsilon = \frac{1}{N} \cdot \sum_{N} \frac{|y - \bar{y}|}{y} \tag{3.17}$$

where y is the ground truth, $\bar{y}$ is the estimated count and N the number of images in the test dataset, is being displayed in Table 3.1

Approach	**MDE**
Ours	0.094
Method presented in [144]	0.770
Method presented in [31]	0.230

Table 3.1 Mean Deviation Error for Counting

As expected from the resulted density images the mean absolute relative error of the two competitive methods is quite high. In the case of [31] the linear regression is unable to learn the proper relationship between density and the count number. Even the approach in

[144], which estimates the count by summing up all the responses from the density map, the counting error is significant.

The counting error can further be reduced, by combining temporal information to remove noise from the measurements. Specifically, the combination of three pipelines with input frames at *t-1*, *t* and *t+1* in order to estimate the count of frame *t* generates a mean relative error of 0.091. Table 3.2 presents results obtained by combining information from varying number of frames (one pipeline per frame is used) using the MDE, the Mean Square Error (MSE) and the Mean Absolute Error (MAE). Considering the framerate of the MALL dataset (2fps) coherence is assumed in a temporal window of 1 sec. For videos with higher framerate, one would expect that optimal performance could be achieved by using more frames.

Number of pipelines	**MAE**	**MSE**	**MDE**
1	3,15	16,9	0,093
3	3,00	15,7	0,091
5	3,77	23,8	0,109
7	5,91	46,6	0,200

Table 3.2: Comparison of varying number of pipelines, where one pipeline per frame is used.

Table 3.3 compares our method with other non-CNN approaches for people counting performed in the MALL dataset, using the MAE the MSE and the MDE. Our approach seems to perform similarly with other people counting methods.

Method	**MAE**	**MSE**	**MDE**
CHEN_1 [25]	3,59	19,0	0,110
CHEN_2 [26]	3,43	17,7	0,105
LOY [86]	-	17,8	-
ZHANG [145]	2,69	12,1	0,082
KUMAGAI [76]	2,89	13,4	0,091
PHAM [109]	2,50	10,0	0,080
OURS	3,00	15.7	0,091

Table 3.3: Comparison with other non-CNN methods in the Mall dataset

3.5 Conclusion

In this chapter a methodology using CNN was presented for people counting. We have demonstrated that using the whole image information as training input instead of using cropped images, generates better results and the network is able to learn better how to distinguish between the foreground and the background. Furthermore by fusing the count estimate in the temporal domain, we can further improve the measurement provided. In the best to our knowledge, our method is the first to propose the application of a CNN on the whole image for the task of people counting and furthermore to use temporal information for the same task. However further investigation is needed in order to improve the accuracy of our estimate.

Similar to chapter 2, this chapter proposed a solution for measuring pedestrian/crowd characteristics. The information gained from such systems can be used as input and not only for pedestrian simulation software. The following chapter by presenting a simulation framework gives the insights to the reader for how such measurements can be used.

CHAPTER FOUR

4. Modeling Pedestrian Shopping Behaviour

4.1 Introduction

In the previous chapter a method for people counting was presented. People counting can be used in a pedestrian simulation context in order to provide measurements regarding the rate of people entering and exiting an environment and also they can be used to measure the footfall of pedestrians in shopping areas which can be used as ground truth in order to validate a pedestrian shopping behavioural model.

A pedestrian shopping model which answers to the question of how a pedestrian decides which shop to visit and which street segment he chooses to explore is of significant importance for urban planners who want to create a more efficient and smart environment, but also for shopping mall managers who want to increase the time a pedestrian spends inside their mall. The performance and the utilisation of an urban pedestrianised space can be analysed with the use of footfall analytics. That is the amount of pedestrians passing through the various street segments of an environment. Thus, in order to be able to answer to the previous questions, the knowledge of the footfall must be known or predicted. It can be argued that the footfall in already established environments can be measured directly by observation. However when the environment is not being manifested, for example it can be in a conceptual phase, we should be able as well to predict the footfall of the pedestrians. So the main question is: "how can we estimate the footfall of pedestrians in a specific real or virtual environment?"

Although most of the approaches trying to model and simulate a pedestrian's shopping behaviour are focused on the pedestrian's shopping agenda or the topological arrangement of the space, we believe that the pedestrian shopping behaviour is dynamic, originated from the pedestrian's perception of the environment. A pedestrian moving into a space occupied with buildings, perceives the environment visually. Low level information (i.e. edges, corners etc) allows the pedestrian to judge the relationship between the various

structures and provides cues for space to be explored. Moreover a fully conscious pedestrian, i.e. a pedestrian who perceives the environment without any liking (e.g. which building is more attractive, which colour is more pleasant), is bound to decisions that utilise this low level information. Thus we propose a novel approach on modelling pedestrians shopping behaviour and estimate their footfall, by assuming that the pedestrians decisions are based on how they perceive the environment in terms of visibility and of the spatial relationship of the buildings in their field of view, without any prior knowledge of the environment.

A pedestrian simulation, that is as realistic as possible so it can imitate actual behaviour and predict a realistic outcome, is required. Our hypothesis is that people behaviour is affected by their environment. Moreover the focus of this research is to establish a model of individual behaviour whose driving force is the visual information that a pedestrian perceives and its high level interpretation of the surrounding environment. Therefore, the work in this chapter will try and simulate people's behaviour in a microscopic level and thus model each pedestrian with an intelligent agent. An intelligent agent is an autonomous entity that perceives its environment through sensors and acts on it through actuators in order to satisfy its inner desires or achieve specific goals [115].

Characteristics such as a pedestrian's preferred speed and the density of people in an environment, as measured in chapter 2 and chapter 3 respectively, must be known in order to create a realistic simulation approach. The former is necessary as it is part of the coherent characterization of the movement of an agent, while the latter can provide with origin/destination measures, which is the amount of people starting/ending from/to a point. Moreover density estimation of people can be used for online calibration of a simulation by adjusting the simulated phenomena to the real measure.

The main contribution of this chapter is the creation of a baseline memory-free cognitive framework for simulating pedestrian shopping behaviour, with the route choices of people based on their visual perception which can be extended in the future to encompass spatial knowledge in complex built environments. Moreover we propose a novel algorithm for calculating the isovist [9] is presented. In Section 4.2 a literature review of modelling pedestrian behaviour and more specifically in shopping behaviour is presented, while in

Section 4.3 the methodology of our approached is explained. Finally in Section 4.4 the experimental results are presented, followed by Section 4.5 which concludes this chapter.

4.2 Previous Work

The study of human pedestrian behaviour is a topic of great interest to many research fields. In social psychology pedestrian behaviour is examined in order to evaluate transport and traffic risks [114][51][52][41][129]. More specifically pedestrian behaviour is analysed and characteristics of the pedestrians, such as age, gender, psychological profiles, are being examined in order to understand factors that increase risk taking and danger perception in pedestrian behaviour.

In computer animation and computer games, pedestrian behaviour is simulated in order to create believable autonomous agents and crowds in a virtual environment [95][122][104][53][101][58][72]. Two are the main approaches on crowd pedestrian simulation: First, methods such as in [95][101][122] attempt to generate a behavioural model for each separate person-agent where layers of rules activated by simulation events drive this agent. Second, a crowd is directly simulated by agents following flocking behaviour [58] or being part of a particle system driven by some physical forces [53][72]. These approaches of modelling pedestrian behaviour do not attempt to evaluate quantitatively their findings; instead they aim to generate systems that seem to imitate apparent crowd behaviours.

In computer vision prior knowledge of pedestrian behaviour is used in surveillance systems in order to improve tracking or detection results [90][15][3][79][27][106][32][138]. Models, such as discreet choice models (DCM)[8], linear trajectory avoidance (LTA)[105], and generally models which try to minimise various energy functions, representing social factors, which are used to simulate pedestrian movement can assist and limit the search space for tracking estimations. Moreover the social force model [59][60][77] can be used as detector of abnormal behaviours on energy estimated maps from particle movements in a video sequence.

Computer vision systems have also been used to extract pedestrian behavioural characteristics. Estimated trajectories of pedestrians have been used to optimise the social

force model parameters which then can be used to simulate pedestrian movement on a microscopic simulation[68]. Tracking and heel localisation of pedestrians has been used to generate speed profiles that in their turn are used for modelling the speed preference of pedestrian in a microscopic simulation [124]. In [117] a pedestrian route choice model is learned by using observed pedestrian trajectories. The trajectories are used for the parameter optimisation of a linear function which is made from the combination of various energy functions which represent pedestrian movement preferences.

Finally in urban planning and transport management, pedestrian behaviour is being modelled with respect to the environment so as to evaluate the design of a space in terms of its usability, to model crowd and evacuation dynamics as well as for commercial activity organisation.

Specifically, in order to model and simulate pedestrian behaviour the itinerary of activities of a pedestrian as well as the interaction between the pedestrian and the environment must be taken under consideration [100]. Moreover pedestrians' actions are dynamic and they are influenced by the presence of other pedestrians. Based on the scale of the simulation but also on the context, macroscopic [34][64] or microscopic models [2][36][74][63][46] can be applied. Macroscopic models consider the environment as a whole and are concerned with representing the aggregate pedestrian behaviour while they do not deal with the underlying dynamics. On the contrary microscopic models, either by using cellular automata or by representing each pedestrian as an individual agent, are able to express the dynamic behaviour and how this is affected by other pedestrians and the environment.

The behaviour of pedestrians varies based on the type of environment they are walking in. For example a person walking in a busy street in London while commuting to work behaves differently from someone who is walking purposelessly in a shopping mall on a Sunday morning. Moreover the same person walking in the shopping mall will behave differently depending on its inner state (i.e. beliefs, purpose of visit), the knowledge of the space, the various constraints (e.g. waiting in a queue), or forced behaviour that maybe imposed (e.g. evacuation) [25]. Thus a unified approach to model pedestrian behaviour in all various contexts may be too complex and therefore infeasible.

The behaviour of pedestrians in multiple attractor environments such as town centres and shopping malls is not driven by an easily identifiable set of objectives and destinations, and crowd dynamics are not the only source of interaction between pedestrians. In these environments there is a multitude of potential activities, various situated attractors and potentially parallel and conflicting objectives. In these environments it is arguably impossible to capture pedestrian behaviour using behavioural models based on sets of variables and rules that adhere to the deterministic paradigm and therefore the emerging route patterns cannot be predicted.

4.2.1 Pedestrian Shopping Approaches

There have been various microscopic approaches that try to grasp the underlying dynamics of pedestrian behaviour in shopping environments. In [143] a method is presented to simulate pedestrian route choices in shopping mall corridors. Their aim is to test the hypothesis that the aggregate pattern is a result of visitors operating according to four simple movement heuristics. The first heuristic deals with creating an itinerary for each pedestrian in the form of a random walk. The second heuristic deals with preserving a minimum length of that walk. The third heuristic is used to represent the pedestrians who are familiar with the shopping environment and have a global perception of the whole area. This heuristic gives higher preference to movement into corridors of the malls that are central and have high connectivity with the rest of the corridors. Finally the fourth heuristic deals with pedestrians visiting the shopping mall with a specific goal. For these the shortest route to a randomly selected store is used in the model. Although the last two heuristics show a high correlation with the actual observed route choices, the model does not take into consideration the attractiveness of different areas of the environment for each agent, dealing with it as a passive network of paths and thus lacking to infuse any dynamic element that affects people choices.

In [37], a framework for processing agent based simulations in shopping environments is presented. The environment is represented as a network of shops and streets, using GIS shape files, where the agents are moving within. Each agent has its own activity agenda comprising of tasks to complete during the shopping trip and an initial path route with the shops to visit. Each agent is equipped with a stochastic perceptual field, informing them of

the immediate surrounding environment in terms of which shops/activities are nearby and accessible to the agent, but with no information regarding the topography of the space. Adding a store in the perceptual field of an agent is based on the agent's motivation, the agent's age, if the agent is part of a group, the store attractiveness, store awareness and store characteristics. At each step of the simulation, the agent has to decide the next move based on what is perceived. The agent might alter its activity agenda by visiting stores that were not included in it, by completing a task (e.g. make a purchase) and by creating new tasks that are induced by other activities. Finally an agent might not complete his activity agenda due to time constraints. The simulation is run on NetLogo [142] which is a programmable modelling environment for simulating social and natural phenomena.

Zu et al [146] presented a method based on bounded rationality. Bounded rationality proposed the idea that decision making of individuals is based on the limited information they possess as well as the amount of time they have to make a decision. Thus by using bounded rationality the decision that a pedestrian will make is not always the optimal. Zu et al argue that the behaviour of a pedestrian in a complex environment such as a shopping centre cannot be explained by utility maximization (i.e. rational) approaches. A multi-agent system is created where each agent's behaviour is modelled using four heuristic models. These models give answer to the decisions a pedestrian has to take during his simulated visit. A pedestrian decides whether or not to leave a shop or to go home, which direction to follow, to take a rest or to visit a shop. Each decision is based on different factors and for each factor a threshold value is modelled using a probability distribution to explain the variability of the individual shopping trips. Specifically, the amount of time spent in the environment is taken into account for deciding whether pedestrians will continue their shopping trip or not. In addition, three factors are considered to specify the direction followed after exiting a shop or after a rest: the previous direction that the pedestrian walked from, the total floor space of the visited shop as well as the total length of the available pedestrianised street towards each candidate direction. Finally a pedestrian decides to take a rest or not based on how much time has spent since the last rest. The result of a decision is based on the combination of its factor values compared to the corresponding thresholds.

Borgers et al [16] presented a method that aims at simulating pedestrian behaviour in shopping streets, including visiting shops. Shopping streets are superimposed with a network of links and at each point a pedestrian occupying a link can chose to move only to an adjacent link. The probability of a link being chosen is modelled using a multinomial logit model and depends on the utility value of that link. The utility value depends on factors such as the location of the link on the street, the direction of the link compared with the direction to the exit, the distance of the link from a shop, the size of a shop, and finally if a shop has been visited before. Shops too are represented as links having their own utility and the time a pedestrian spends in a shop is correlated with the size of the shop.

In [74] a multi-agent simulation is used to explore the shortest-path rule and utility maximization of pedestrians in shopping areas. Each agent has their own activity agenda and at each cycle the agent considers the next move based on four different processes. During the first process the agent gathers information about the environment such as information regarding the shops in his agenda, the street network and other agents. This information is then compared with the information gathered in previous cycles. In the second process marketing data in the form of shop spatial data is used with a neural network algorithm along with the agent preferences. This provides in the form of a probability an attractiveness score for each shop to the agent. In the third process the optimal route is chosen using a mixed logit model as the basis of the optimization. This can be further refined by introducing travelling salesman algorithms. Finally in the fourth process a collision avoidance model is employed. During the simulation Genetic Algorithms are employed in order to seek the parameters and optimal solution for the maximization of utility function built in the model.

The model we present here is different from the aforementioned approaches because it links the information model that drives pedestrian behaviour with spatial configuration in the sense that all routing and shopping decisions are based on visual perception. Using visual perception allow us to infuse real world information in the agent decision mechanism and thus take into consideration how things actual appear. How large are some streets, how much of the perceptual field of a pedestrian occupies a shop, these questions are answered dynamically by the association of a pedestrian at a specific location with the environment. At the same time our model remains decisively activity-driven because it

recognises the importance of the location of shops and other activities and these shape agent behaviour in an explicit and direct manner. As such, it represents a hybrid between activity-driven models (that fail to recognise the role of spatial configuration in shopping behaviour)[146] and spatial analysis models [133][143], which focus mainly on the topological and geometrical attributes of the environment.

4.3 Methodology

The proposed methodology attempts to model pedestrian shopping behaviour, based on how the environment is visually perceived by pedestrian agents and therefore affects their decisions. Simulation is being used as a tool, however the focus is on the perception of the environment by the pedestrian and its effect on his behaviour.

As mentioned in Section 4.1, an intelligent agent perceives its environment through sensors and acts on it through actuators. Thus in order to simulate the shopping behaviour of a pedestrian, represented by an intelligent agent, the parameters that need to be considered are: the environment in which the shopping agent exists, the state of the agent at any given time, the sensors and actuators of the agent, and finally the agent's internal rationale of translating the environmental stimuli into actions. In real life most visitors in a shopping area commonly have some prior knowledge of the topology of the space and the position of each shop in it. However, in the scope of this chapter, it is assumed that the agents in the simulation have no prior knowledge of the environment. We are interested in finding if the configuration of an environment, populated with environmental agnostic agents, can by itself provide visual cues to the pedestrians that will match the ground truth data. Thus our aim is to create an initial framework in order to simulate pedestrian shopping behaviour which can be extended in the future so as to encompass this prior knowledge in a more complex model.

4.3.1 Environment

The environment where the agent moves is stored in GIS shape files. These files contain geographical information represented through vector data and describe the geometry of the environment. Thus the various shops and streets are represented using polygons and lines in a two dimensional space. An example of a shape file is presented in Figure 4.1.

A shape file contains a set Λ of line segments where $\Lambda = \{\lambda_w\}, w = 1..N_\lambda$. In the environment a set of shops S exists where $S = \{s_1, s_2, \ldots, s_{N_s}\}$ where N_s is the total number of shops. Each shop is defined by a polygon p_s with area q_s, type of shop w_s and value v_s. Each polygon has one or more lines $l_{s,r}$ indicating the *r-th* entrance of the shop. The environment contains a set of entry lines E where the agents are created for the simulation and a set of exit lines E′ where the agents exit the simulation.

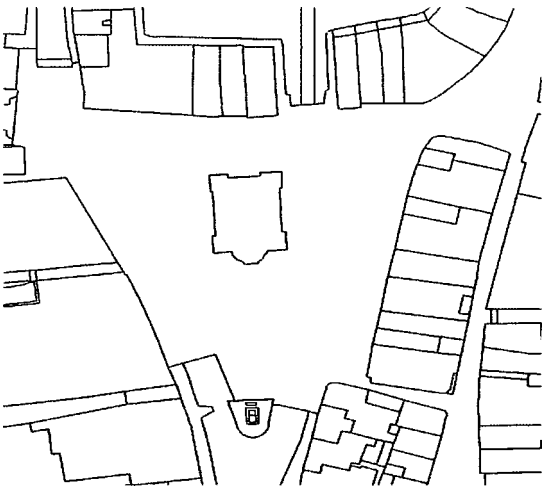


Figure 4.1 : Shape file displaying Kingston's Market Square

4.3.2 Agent

Each agent i where $i = 1,2, \ldots, N_i$ with N_i the total number of agents at any given time in the simulation, is characterised by its location in the environment, given by the coordinates x_i and y_i, the unit vector indicating the direction that he/she faces $\overrightarrow{d_i}$, its speed preference V_{i0}, the remaining time in seconds for its shopping trip t_i, and a set of activities I_i that defines the itinerary of the agent. Specifically, I_i contains the different types of shops that the agent needs to visit during its shopping trip. Each agent perceives the environment through its vision. More specifically, based on the agent's location and direction, an agent has a $\hat{\phi}$ degree forward facing field of view of length d with $\overrightarrow{d_i}$ acting as a bisector of that field (Figure 4.2).

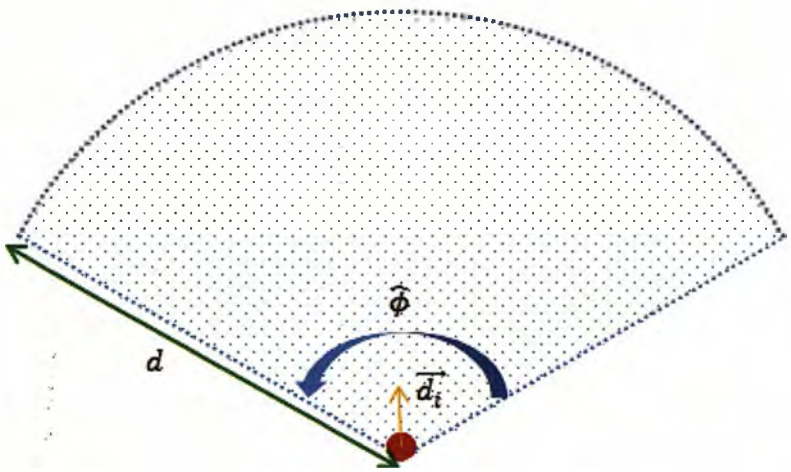


Figure 4.2: Agent's field of view: The agent represented as a red circle has facing direction $\vec{d_i}$, with a field of view of angle $\vec{\phi}$ and length d

4.3.2.1 Agent's Vision

The vision of the agent is the most important aspect of the simulation since his decisions are based on what he sees in the space while moving. In other words, an agent is a conscious entity, being fully aware of his environment and taking decisions based on its perceived configuration. Agent's vision is based on the calculation of the isovist [9] on his current location. An isovist is the set of all points visible from a given vantage point in an environment and an example of an isovist in a 2D space is displayed in Figure 4.3.

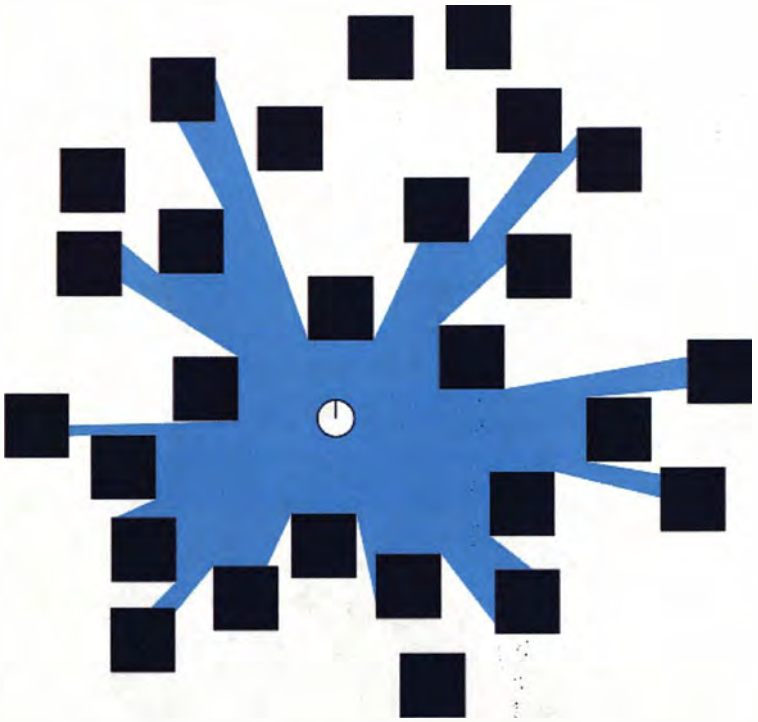


Figure 4.3: The polygon with the light blue colour defines the isovist from a pedestrian position (white circle). The dark blue rectangles are obstacles, obstructing the pedestrian's view. Image adopted from [141].

The most common way of calculating the isovist is by using ray casting. Using this technique, rays from the point of interest are casted towards all directions and their points of intersection with the environment are identified. Then the isovist is calculated as the polygon defined by the nearest points of ray intersection with obstacles. This technique is computationally heavy because the number of rays that need to be cast is large enough and therefore the points of intersections that have to be calculated are proportionally large. For example if a ray is being cast for every angular degree of a 180° field of view, then the intersections of all 180 rays with the environment must be calculated. Moreover the number of rays being cast defines the resolution of the environment being perceived and therefore, high precision requires a large number of cast rays. In order to reduce the computational burden and be precise in the isovist calculation we have developed a novel algorithm that uses information regarding the topology of the space. For small environments, where the number of corners present is low, our algorithm performs faster than the ray casting, however as the complexity of the environment increases the complexity of our algorithm increases as well, while ray casting computational cost is

constant regardless the complexity of the space. The steps of the algorithm for calculating the isovist from a point in the environment can be summarised in the flowchart as displayed in Figure 4.4.

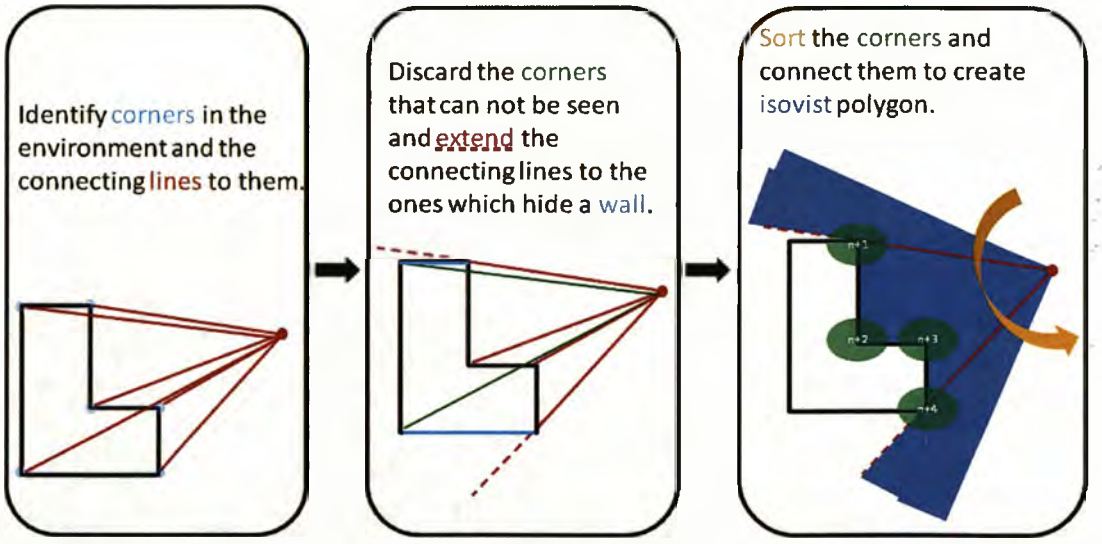


Figure 4.4: Flowchart of isovist calculation algorithm

Initially a grid G is overlaid on the space of the environment, and for each cell c_q of the grid, where $q = 1,2,\ldots,N_c$, with N_c the total number of cells, the line segments $\{\lambda_k\}$ that pass from a cell are recorded (Figure 4.5).

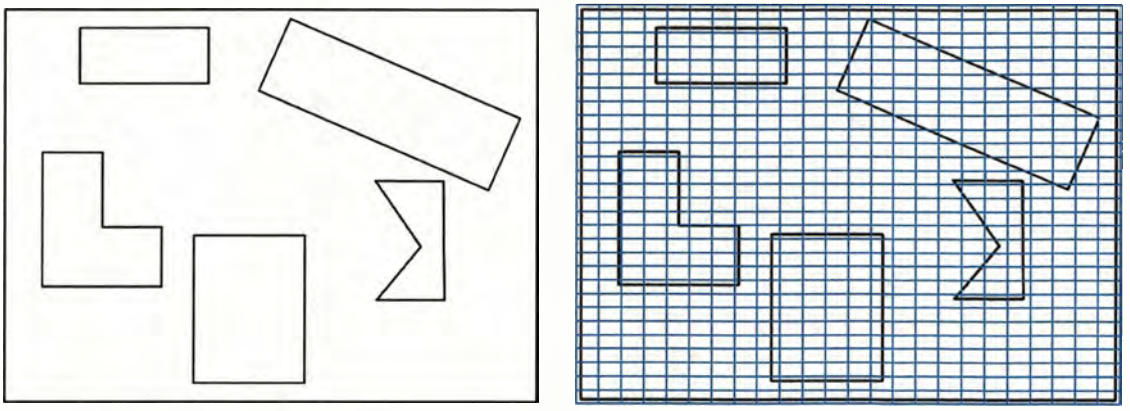
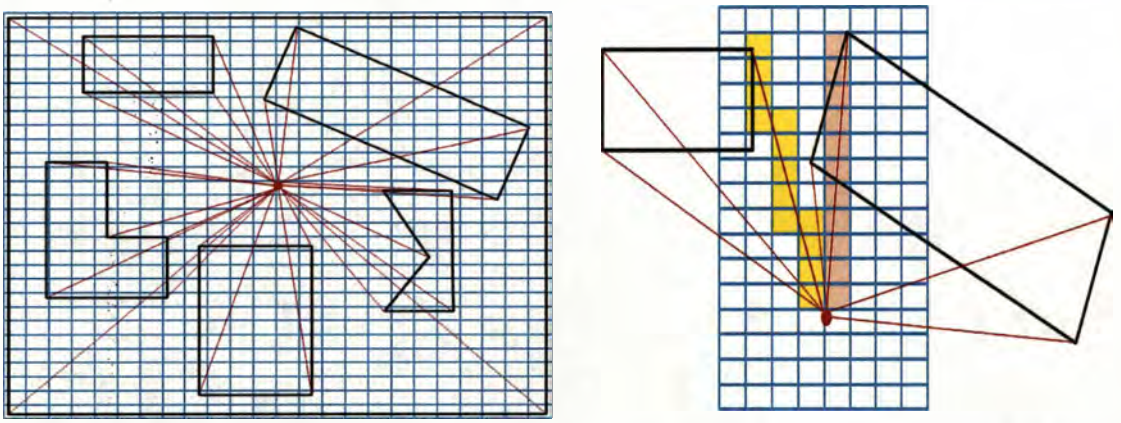


Figure 4.5: On the left the 2D spatial arrangement of the environment. In the right image the grid *G* is overlaid on it.

Assuming an agent i is present in the environment, the set of vertices $M = \{m_1, m_{2,}, \dots, m_{N_m}\}$ of the layout that can be directly seen from the agent's position (x_i, y_i) are determined. In order to achieve this, we compute the set of $\{ \beta^{\delta}_{x_i,y_i} \}$ line segments from (x_i, y_i) to all the vertices present in the environment where $\delta = 1,2, \dots, N_m$ with N_m the number of vertices present in the layout (Figure 4.6(a)). Then, by using an extended version of Bresenham's [21] line algorithm, each line $\beta^{\delta}_{x_i,y_i}$ associated with the various grid cells, $C_{x_i,y_i,\delta} = \{c_q\}$, that crosses (Figure 4.6(b)). The original Bresenham's line algorithm determines which grid cells are approximating a straight line between two given points, while its extension estimates all the grid cells that the line crosses.



(a) (b)

Figure 4.6: (a) Lines from the agent's position to all the present vertices in the layout, (b) the grid cells associated with two of the lines.

Then for each cell, the relationship between the recorded line segments $\{\lambda_w\}$ of the environment and the lines-of-sight $\{\beta^{\delta}_{x_i,y_i}\}$ is examined. If a line-of-sight $\beta^{\delta}_{x_i,y_i}$ crosses a line-segment λ_w then it is disregarded, as the corresponding vertex of this line $\beta^{\delta}_{x_i,y_i}$ is not visible to the agent. For example in Figure 4.6(a) the line that connects the agent with the top right vertex of the enclosing parallelogram is discarded as it crosses two lines in order to reach that vertex. On the other side, if a line-of-sight $\beta^{\delta}_{x_i,y_i}$ only "touches" a line segment λ_w at the point of the vertex, the line-of-sight is kept for the next step of the algorithm (Figure 4.7).

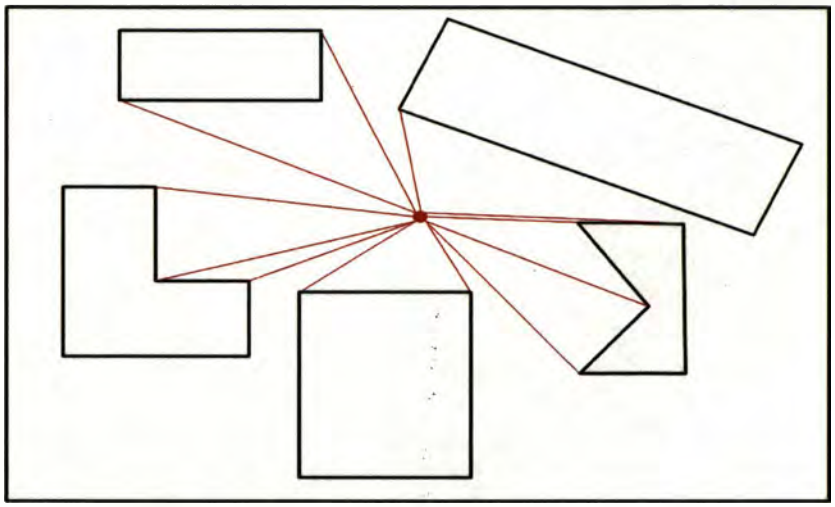


Figure 4.7: The lines-of-sight $\beta^{\delta}_{x_i,y_i}$ that "touch" line segments λ_w at the vertex points.

In the next step we examine the relationship of each remaining $\beta^{\delta}_{x_i,y_i}$ with the line segments connected to its vertex m_w. (Figure 4.8). Each $\beta^{\delta}_{x_i,y_i}$ can be considered that it divides the space into two surfaces. If both edges attached to the vertex m_w are present in the same surface, then this denotes that $\beta^{\delta}_{x_i,y_i}$ can be further extended. Otherwise, if the edges attached to $\beta^{\delta}_{x_i,y_i}$ are laying on different semi-surfaces then this means that $\beta^{\delta}_{x_i,y_i}$ cannot be further extended.

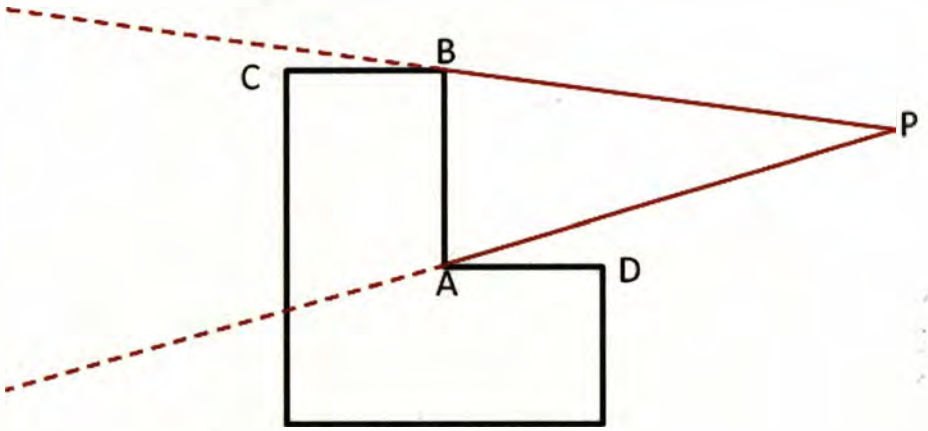


Figure 4.8: Line segment PB can be further extended as BC and BA lie on the same semi-surface. Line segment PA canot be further extended as AB and AD lie on different semi-surfaces.

After extending the line segments, and by using again the extended Bresenham's line algorithm we compute the points where the extended $\beta^{\delta}_{x_i,y_i}$ cross the line segments of the layout, and along with the points of the vertices computed before they formulate the vertices of the isovist polygon (Figure 4.11).

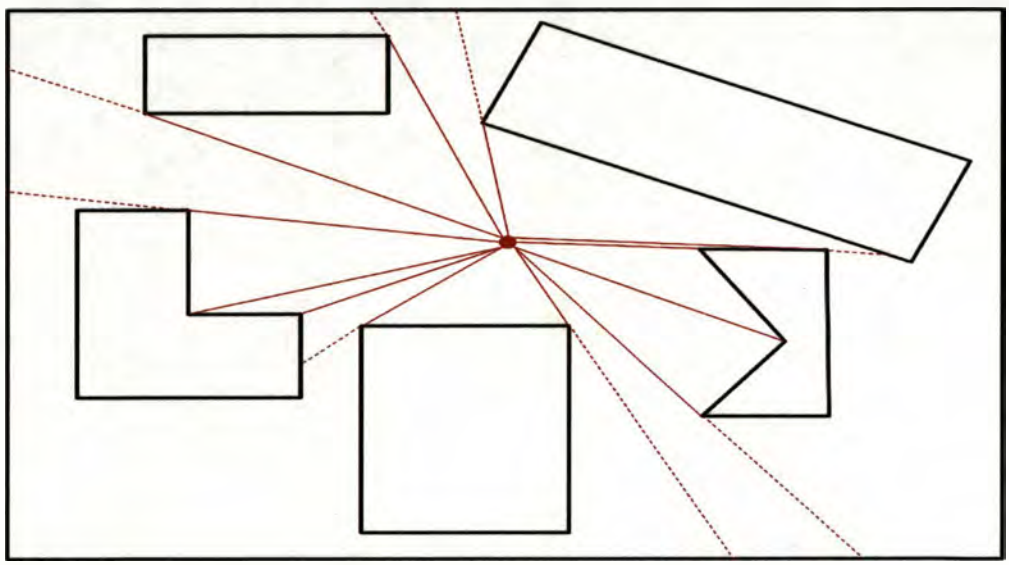


Figure 4.9: Dotted lines represent the extensions of $\beta^{\delta}_{x_i,y_i}$

In order to identify the sequence of the vertices and thus define the space of the layout that the polygon occupies, we initially sort the vertices according to the counter-clockwise angle they form between $\overrightarrow{d_i}$ and the line that connects (x_i,y_i) and m_w. We also compute the distance of the polygon vertices from (x_i,y_i). The vertices of the polygon that were computed from the extensions of $\beta^{\delta}_{x_i,y_i}$, will have the same angle as the corresponding vertices computed before the extension. For instance, both R (original vertex point before extension) and Q (intersection point after extension) in Figure 4.10 have the same angle with $\overrightarrow{d_i}$. Thus, to identify the order of the vertices and to define the isovist polygon, we once more examine the relationship of the $\beta^{\delta}_{x_i,y_i}$ with the edges of vertex m_w., in counter clockwise order. If both edges of vertex m_w lay on the right side (clockwise) of $\beta^{\delta}_{x_i,y_i}$ (e.g. in Figure 4.10 both edges of vertex B lay on the right side of PB) then, vertex m_w is added to the polygon vertex sequence before the point identified by the extension of $\beta^{\delta}_{x_i,y_i}$. Otherwise, if both edges of vertex m_w lay on the left side (counter clockwise) of $\beta^{\delta}_{x_i,y_i}$ then the point identified from the extension of $\beta^{\delta}_{x_i,y_i}$ is added first. Note that if one edge lays on the right side of m_w while the other on the left then there is no extension of $\beta^{\delta}_{x_i,y_i}$to be considered and thus only the m_w is added directly to the polygon vertex sequence. An example of applying the sorting algorithm can be viewed in Figure 4.10, where starting from vertex A, we examine the edges of vertex B. Since both edges lay on the right side of PB then the point B is added to the isovist polygon vertex sequence, followed by point C. In the case of point R, both of its edges lay on the left side of PR thus the point Q is added in the sequence before R.

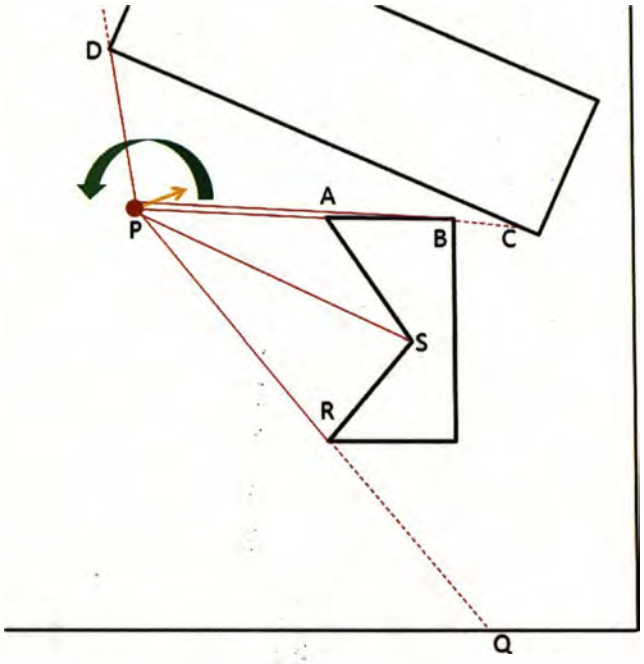


Figure 4.10: Sorted isovist polygon vertices. The green arrow shows the direction (counter-clockwise) the sorting algorithm is being applied, while the orange arrow indicated the facing direction of the agent ($\vec{d_t}$).

Finally, after sorting the vertices, the isovist polygon is computed (Figure 4.11).

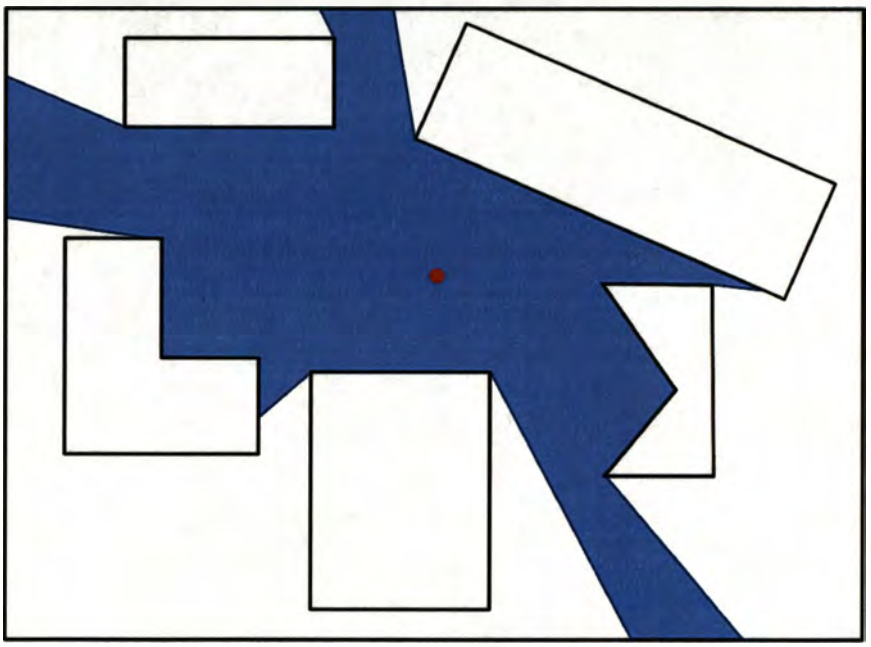


Figure 4.11: The calculated isovist polygon (blue area) of an agent (red circle)

Assuming two line segments, v_1 and v_2 with length d and starting from (x_i,y_i) with

$$\widehat{v_1\vec{d_i}} = \widehat{\vec{d_i}v_2} \text{ and } \widehat{v_1\vec{d_i}} + \widehat{\vec{d_i}v_2} = \hat{\varphi}$$

the isovist polygon is split in to two parts. The part of the isovist which includes $\vec{d_i}$ defines the polygon V_i^t which represents the area of the visual perception of agent i at time t. In Figure 4.12 the isovist polygon is displayed where $\hat{\varphi}$ is 180 degrees.

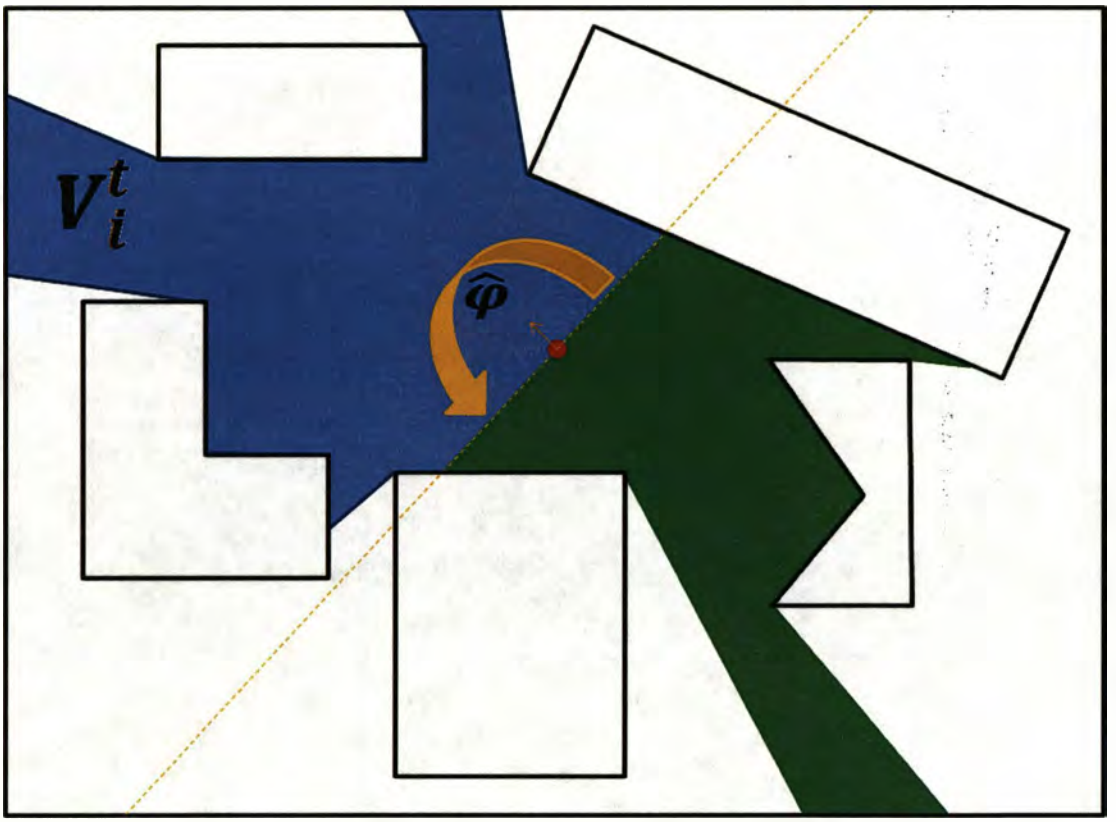


Figure 4.12: Separation of isovist. A pedestrian having a direction of movement denoted with the orange vector, views the part of the isovist polygon V_i^t, that is constrained by his field of view angle $\hat{\varphi}$ ($\hat{\varphi}$ = 180° here), indicated by blue colour.

The pedestrian moves in a 2D environment, thus what he actually "perceives" through vision are line segments which represent the walls of the various buildings, shop entrances, the environment boundaries and the space that exists between these segments. Thus at each moment t the visual information of an agent i is the counter-clockwise arranged set S_i^t of

line segments which occurs from the intersection of V_i^t with the environment. Hence the pedestrian in Figure 4.12 perceives the line segments that are displayed in Figure 4.13.

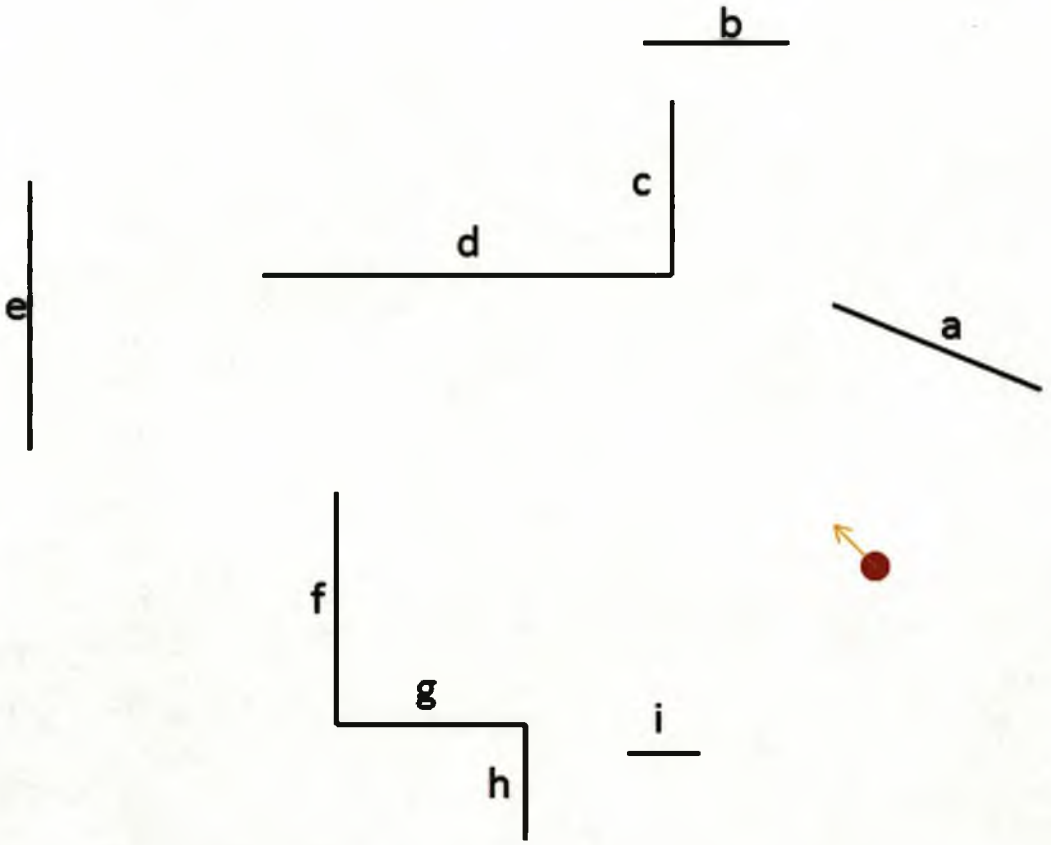


Figure 4.13: Line segments perceived by an agent (red circle) facing the environment according to yellow arrow.

4.3.2.2 Agent's Movement

An agent moves within the environment following the Social Force phenomenological model [59][60][77]. This model was created in order to simulate collective behaviour of pedestrians moving within an environment and assumes that people are under the constant influence of various forces. These forces are either generated by external factors such as the existence of other pedestrians or structures in the close neighbourhood of a pedestrian, or by internal desires of the pedestrian such as its preferred speed and its target. More precisely the social force model can analytically be described as following. The force that a pedestrian experiences at any point in its environment is the accumulation

of three different forces. The first one, $\vec{f}_{ij}$ is the force that is applied to pedestrian i in reaction to the existence of pedestrian j. The second one $\vec{f}_{i\lambda}$ is the force that is applied to pedestrian i in response to reaction with a structure λ_k, and finally $\vec{f}_{preferred}$ is the force that is applied "internally" to the pedestrian.

$$\vec{f}_i = \sum_j \vec{f}_{ij} + \sum_{\lambda_k} \vec{f}_{i\lambda_k} + \vec{f}_{preferred} \quad (4.1)$$

The force $\vec{f}_{ij}$ that the pedestrian i experienced due to the presence of pedestrian j is given by Equation 4.2.

$$\vec{f}_{ij} = \vec{f}_{social\ repulsion} + \vec{f}_{pushing} + \vec{f}_{friction} \quad (4.2)$$

Where

$$\vec{f}_{social\ repulsion} = A \cdot e^{(R_{ij}-d_{ij})/B} \cdot \vec{n}_{ij} \quad (4.3)$$

$$\vec{f}_{pushing} = k \cdot \eta \cdot (R_{ij} - d_{ij}) \cdot \vec{n}_{ij} \quad (4.4)$$

$$\vec{f}_{friction} = \kappa \cdot |\vec{f}_{pushing}| \cdot \vec{t}_{ij} \quad (4.5)$$

In the above equations A, B, k and κ are constants. R_{ij} is the sum of the radius of pedestrian i and j, d_{ij} is the distance between pedestrian i and j , $\vec{n}_{ij}$ is the vector pointing from i to j and $\vec{t}_{ij}$ is the vector in the tangential direction and direct opposite to the velocity of pedestrian i. The value of η depends on the distance between two pedestrians, and in the case that the pedestrians do not touch each other it takes a value of 0, thus eliminating the forces of pushing and friction.

Force $\vec{f}_{i\lambda}$ is similar to $\vec{f}_{ij}$ only that it is calculated based on the relation of pedestrian i and a line segment λ representing an obstacle or structure.

Finally

$$\vec{f}_{preferred} = -m_i \cdot \frac{\vec{u}_i - \vec{u}_{i0}}{\tau}\ ,\ \vec{u}_{i0} = (1-p) \cdot V_{i0} \cdot \overrightarrow{d_i} + p \cdot \langle \vec{u}_j \rangle_i \quad (4.6)$$

where m_i and $\vec{u}_i$ are the mass and current velocity of the pedestrian, τ is his reaction time, $\vec{u}_{i0}$ is the preferred velocity, V_{i0} is the speed that the pedestrian would prefer to move, $\overrightarrow{d_i}$ is

the unit vector along the pedestrian's direction and $\langle \vec{u}_j \rangle_i$ is the average velocity the pedestrian i perceives from the pedestrian's moving in his neighbourhood. Finally p is a constant parameter indicating how strongly a pedestrian prefers to move based on its preferred speed, and thus staying unaffected of the moving pedestrians around it.

Our hypothesis is that pedestrians move in the environment in such a way that they will try to maximise their potential for new information. That is, they select where to move, based on their visual information, and trying to maximise their chance to reach large areas. Thus along with the Social Force model we have added two more forces, as shown in Equations 4.10-4.12, that assist the pedestrians to fulfil their desires and which generate the preferred velocity of the pedestrian $\vec{u}_{i0}$ by modifying the direction, $\vec{d_i}$, of the movement of the agent. However these are more related to the exploration that a pedestrian undergoes and for this reason will be explained in more detail in Section 4.3.3.3.

4.3.3 Simulation

During the simulation, the environment is assumed static while the agents' behaviour provides the dynamic component of the simulation. An agent's life in the simulation starts by its creation at an entry point. Then the agent can explore the space by perceiving its environment, enter and exit shops to fulfil its shopping agenda and finally exit the simulation through an exit point. In Figure 4.14 the various states of the agent are displayed showing the dependencies between them. Each simulation step accounts for T seconds of time indicating half gait cycle and at each step the agent's available shopping time t_i is reduced by this amount.

Figure 4.14: Agent's states

4.3.3.1 Entering Environment

An agent i enters the environment at a random entry point ε_ζ of the entry line ζ where $\zeta \in E$. Thus the initial coordinates of the agent are the same as the coordinates of ε_ζ and the agent's direction is perpendicular to ζ. The agent is assigned a speed drawn from a speed profile distribution and this speed denotes the preferred walking speed of the pedestrian for its lifetime during the simulation. For each pedestrian i entering the environment an itinerary I_i is assigned. The size of the itinerary is determined randomly, sampling from a uniform distribution with minimum value 0 (no shop to visit) and maximum value N_i. The itinerary contains the types of shops (Figure 4.24) to be visited and a type of shop is selected based on a uniform distribution of the N_ε number of different shop types. After a pedestrian has entered the environment it tries to identify if any existing shop in his perceived section of the environment and its state changes to 'Searching' (Section 4.3.3.2).

4.3.3.2 Searching

The agent uses its field of view to perceive his environment. What the agent is actually able to perceive, as mentioned before, is the set of line segments produced by the intersection of the isovist polygon with the environment. The agent checks whether each identified shop type is in his itinerary I_i. If none of the types of the shops exist in the agent's itinerary then the agent's mode changes to the exploration state. If only one shop has a type that exists in I_i then the agent's mode changes to the moving to target state.

Otherwise if more than one shop have been identified as possible targets, then the agent judges stochastically which shop is more attractive based on a factor ϱ_s that depends on the distance to the entrance of the shop from the agent, the angle that the shop occupies in the agent's visual field and the shop's area. The attraction of a pedestrian to a shop (Figure 4.15) is based on three factors. The first factor denotes how large is the storefront of a shop ($\sum_r |l_{is,r}|$) compared to the others observed. The second factor ($\widehat{\sum_r \phi_{ts}}$) indicates how dominant a shop is in the agent's visual field and it is measured as the angle formed between the agent's position and the edges of the visible line segments of the shop. The third factor (d_{is}) denotes how close to a shop the agent is, and it is normalised according to the maximum and minimum distance of the agent from the shops entrances. Thus the score ϱ_s for each of the candidate shops s is:

$$\varrho_s = \frac{\sum_r |l_{is,r}| - q_{i,min}}{q_{i,max} - q_{i,min}} + \frac{\widehat{\sum_r \phi_{is}}}{\hat{\phi}} + \frac{d_{i,max} - d_{is}}{d_{i,max} - d_{i,min}} \quad (4.7)$$

where $q_{i,min}, q_{i,max}$ are the smallest and largest storefront widths of candidate shop entrance segments respectively, $d_{i,min}, d_{i,max}$ the smallest and largest distance to the present candidate shops and r is the number of entrance segments for the candidate shop. Whenever a shop has been selected as a target, the closest point of that shop to the agent becomes the target point of the agent.

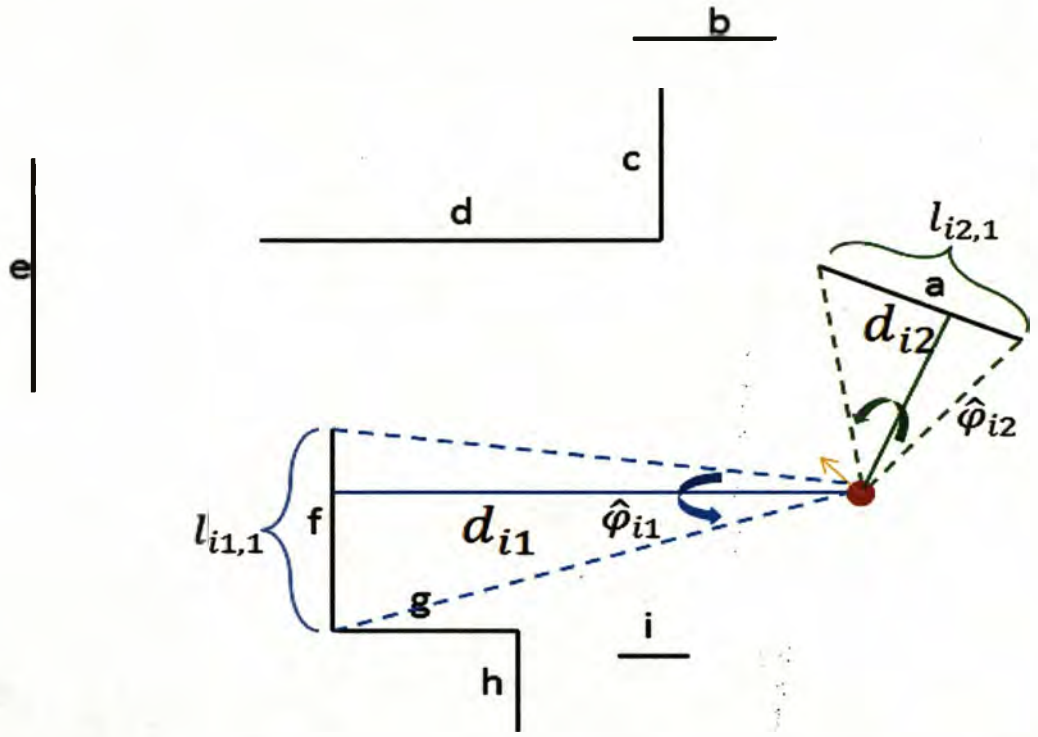


Figure 4.15: Visibility factors of two shops. An agent (red circle) with direction indicated by the yellow arrow will consider the attraction that a shop has to it based on the distances, d_{i1} and d_{i2}, from the shop, the angles, $\hat{\varphi}_{i1}$ and $\hat{\varphi}_{i2}$, that the shop store front occupies in it's visual field and the lengths, $l_{i1,r}$ and $l_{i2,r}$, of the store front.

When an agent's shopping time has depleted or, the agent's itinerary is empty, the agent will be searching for an exit. In this case, the agent tries to identify if any of the exit line segments is within its isovist visual field. If an exit is identified then the agent will select as target a point from the exit line segment and its state changes to 'Moving to Target'(Section 4.3.3.4). The same state transition occurs when a shop that satisfy the agent's needs is identified else his state becomes 'Exploring' (Section 4.3.3.3) in order to find shops that will satisfy its itinerary needs.

4.3.3.3 Exploring

The agent will be in the exploring state when either no suitable shop is within his field of view, or it looks for an exit. When the agent is in that mode, it will try and move to places that have higher potential of new information, i.e. it will try to move to places that are

seemed to lead to large open spaces. This is achieved by identifying the openings o that occur between unconnected consecutive line segments in S_i^t, excluding the opening between the first and last segment. In these openings the larger the distance between the line segments, the higher the potential of a large open space to explore. In Figure 4.16 the red lines specify the distance between two line segments, the larger the distance the higher the potential of more information.

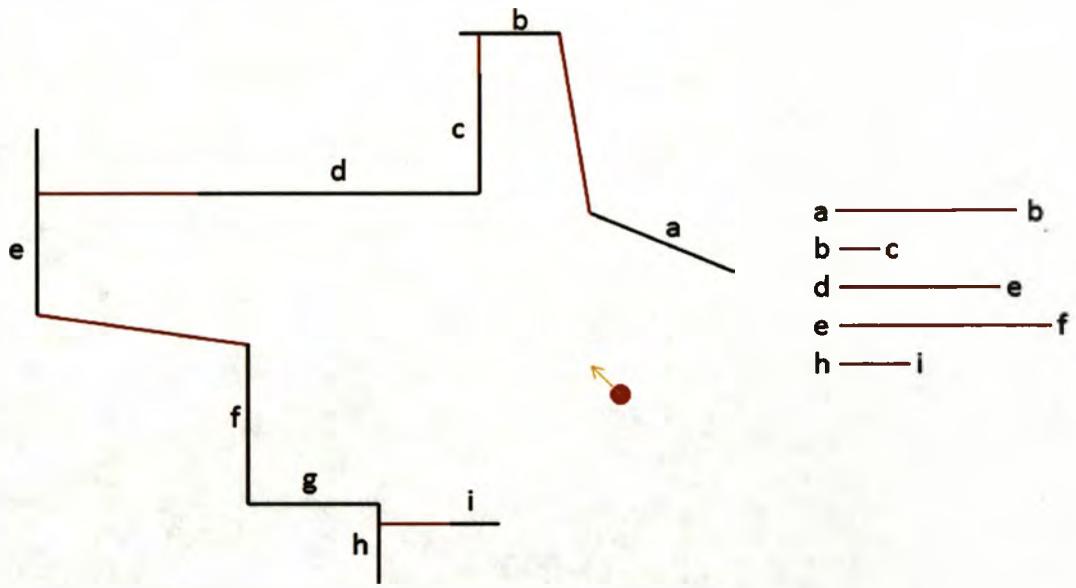


Figure 4.16: Potential of information based on distances (red) between line segments (black). In the right side of the figure, the shortest distance between the line segments, corresponding to the various openings in the environment for exploration, are placed against each other. It can be seen that the distance between e and f line segments is the longest and thus from the five openings for exploration the e-f presents the highest potential.

The arrangement of the line segments in a given urban space is also a good indicator of the size of the hidden area. For example in Figure 4.17, an agent, represented by a red dot with direction indicated by a yellow arrow, will be able to see the black line segments representing the various structures in his environment. We can see the angles, represented by yellow arched arrows, which are formed with consecutive unconnected line segments. As the angle between two line segments gets closer to a right angle the probability that the space, between these two line segments, is richer in information gets higher.

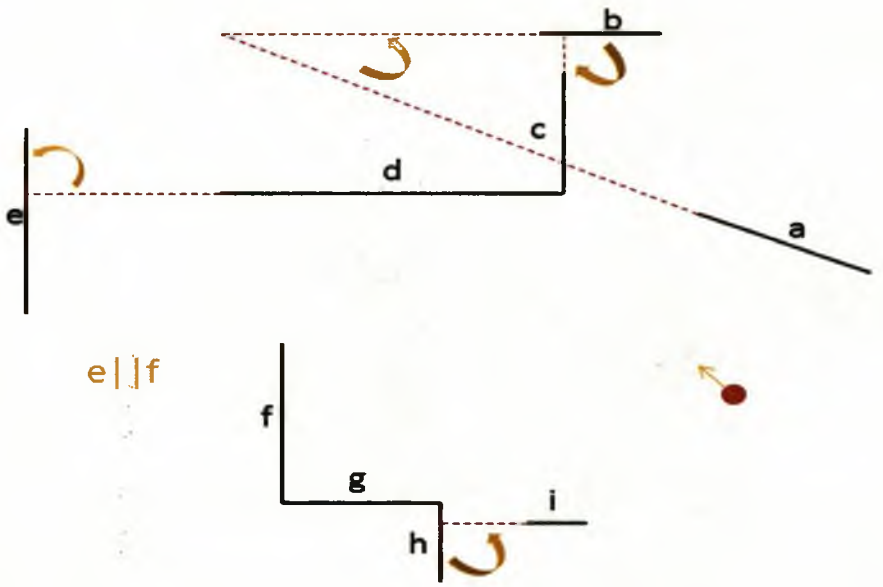


Figure 4.17: Potential of information based on spatial arrangement of visible line segments. In the figure the angles (yellow arrows) between the line segments (black) or their extensions (red dotted lines) that create the openings can be seen. The angles are formed by taking into consideration the counter clockwise arrangement of line segments.

More specifically the agent will assign a score to each of the openings in its visual field. Based on the openings that an agent perceives at a time, the distance which is measured as the smallest distance between the line segments that form this opening is being normalised using the formula

$$d_{o,norm} = \frac{d_{o,actual} - d_{S_i^t,min}}{d_{S_i^t,max} - d_{S_i^t,min}} \quad (4.8)$$

where $d_{o,actual}$ is the distance between two line segments, and $d_{S_i^t,max}$ and $d_{S_i^t,min}$ is the largest and smallest distance respectively between neighbouring line segments in S_i^t. Each opening is also characterised by the sine of the angle between the two line segments that form it. Thus the score of each opening present in the visual perception of an agent is given by

$$score_o = w_d \cdot d_{o,norm} + w_a \cdot \sin(o) \quad (4.9)$$

where w_d and w_a are the weights for the distance and angle characterising the opening respectively. Calculating the scores for all openings at a specific time, a discreet probability is assigned to each one of them and an opening o' is selected using statistical

sampling. The closest point, $p_{o'}$ of the opening to the agent, which can be also identified as the edge of the line segment which is closest towards the other line segment and co-creator of the opening is selected as the next target to move (Figure 4.18).

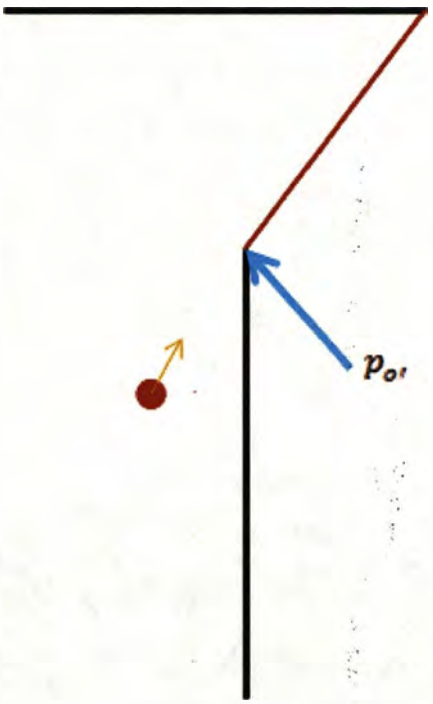


Figure 4.18: A pedestrian (red circle) selects to explore an opening (red line) and therefore its new direction (yellow vector) is pointing towards the next target point $p_{o'}$

As mentioned in Section 4.3.2.2 an agent's preferred velocity while being in the exploratory state will be $\vec{u}_{i0}$. This velocity is affected by the direction, $\overrightarrow{d_i}$, of the movement of the agent. Although the point $p_{o'}$ was selected as the target from the agent, the direction the agent will chose is not a direct straight line towards it. As mentioned before, our assumption is that the agent always tries to explore areas of potentially high information; i.e. the areas that the agent thinks that they have higher potential to be more favourable, as quantified by Equation 4.9. Thus the movement of the agent should not only be energy efficient, but also allow it to examine early the potential of the new unexplored space and therefore allow it to assess its choice. Assuming a circle with centre the point $p_{o'}$ and radius the distance of the agent to that point, the agent experiences two forces (Figure 4.19). The first one, $\overrightarrow{\alpha_1}$, is the attraction that the agent "feels" towards $p_{o'}$ and which force has direction along the radius of the circle. The second force, $\overrightarrow{\alpha_2}$, represents the curiosity of

the pedestrian to see more of where it is going keeping himself the same distance from the target, is along the tangent of the circle and thus be perpendicular to the direction of the attractive force. As the agent moves into a two dimensional space, there are two cases that need to be considered. The first one is when the agent needs to move clockwise in order to explore the opening and the second when he needs to move anti-clockwise. The final force, $\vec{\alpha}$ that an agent with be applied to the agent will be

$$\vec{\alpha} = \overrightarrow{\alpha_1} + \overrightarrow{\alpha_2} \quad (4.10)$$

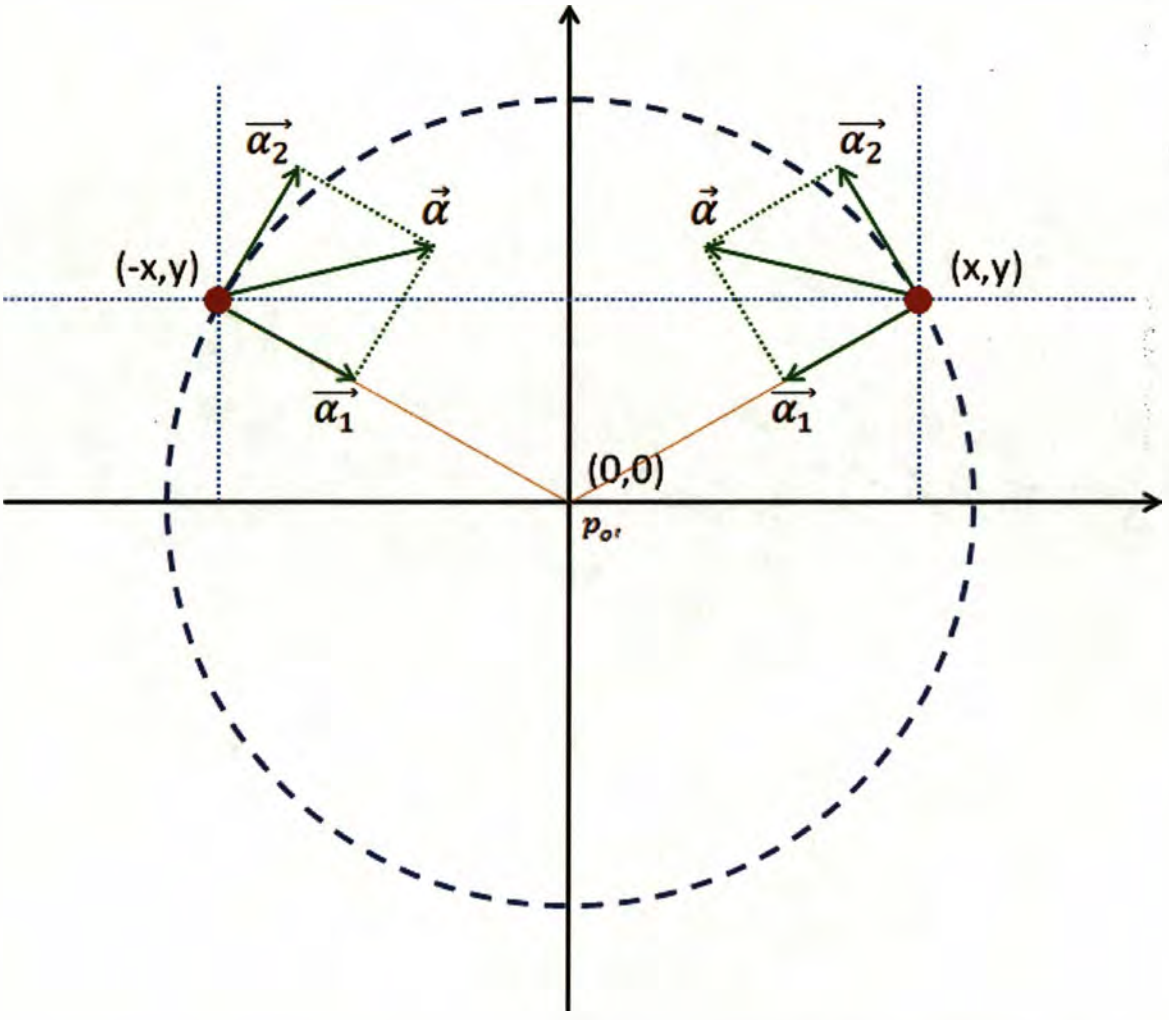


Figure 4.19: Forces two pedestrians (red circles) experience around a selected target (p_{or}). Force $\overrightarrow{\alpha_1}$ is the attractor force towards the target while force $\overrightarrow{\alpha_2}$ represents the curiosity of pedestrian to explore as early as possible the potential of a selected opening. Finally $\vec{\alpha}$ is the combination of the two forces and which describes the pedestrian's path.

where,

$$\vec{a}_1 = w_\varepsilon \cdot \begin{bmatrix} -x \\ -y \end{bmatrix} \quad (4.11)$$

$$\vec{a}_2 = \begin{cases} \frac{wc}{\sqrt{x^2+y^2}} \cdot \begin{bmatrix} y \\ -x \end{bmatrix} \\ \frac{wc}{\sqrt{x^2+y^2}} \cdot \begin{bmatrix} -y \\ x \end{bmatrix} \end{cases} \quad (4.12)$$

with w_c and w_ε being weights.

The magnitude of $\overrightarrow{\alpha_1}$ is affected by the distance from the target point. Thus the attractive force becomes weaker as the agent approaches the target point. The direction of the clockwise $\vec{\alpha}$ in the space when $w_c = w_\varepsilon = 1$, is being displayed in Figure 4.20 using a velocity field, while in Figure 4.21 the counter-clockwise direction of $\overrightarrow{\alpha_1}$ is being shown when $w_c = 2$ and $w_\varepsilon = 1$.

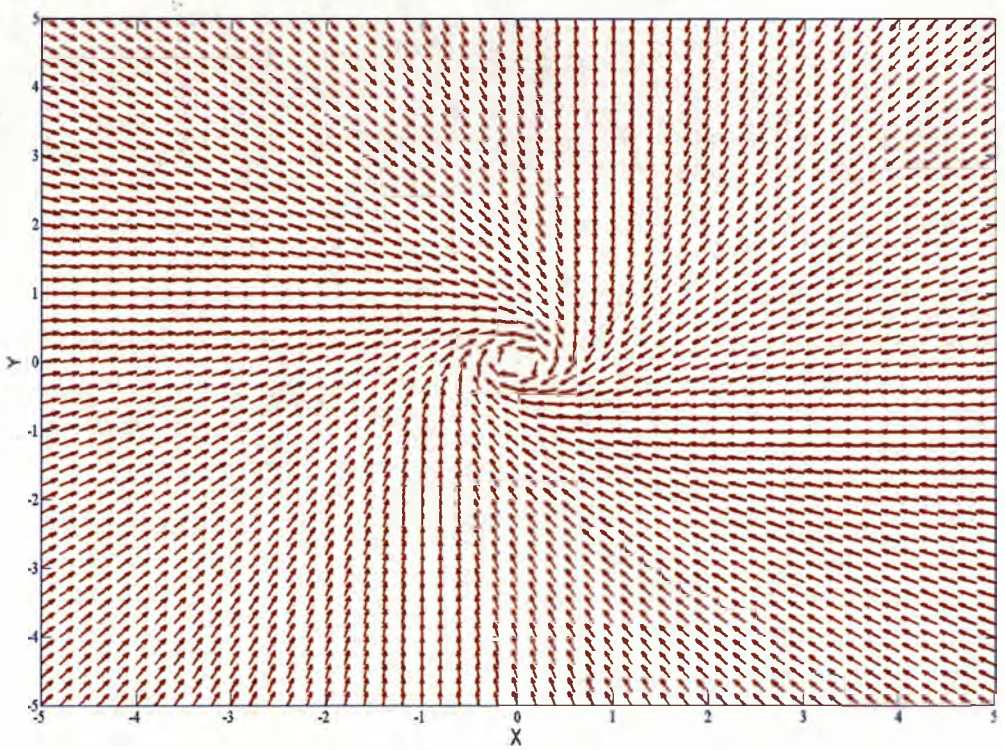


Figure 4.20: Velocity field showing the direction of $\vec{\alpha}$ when agent is turning clockwise.

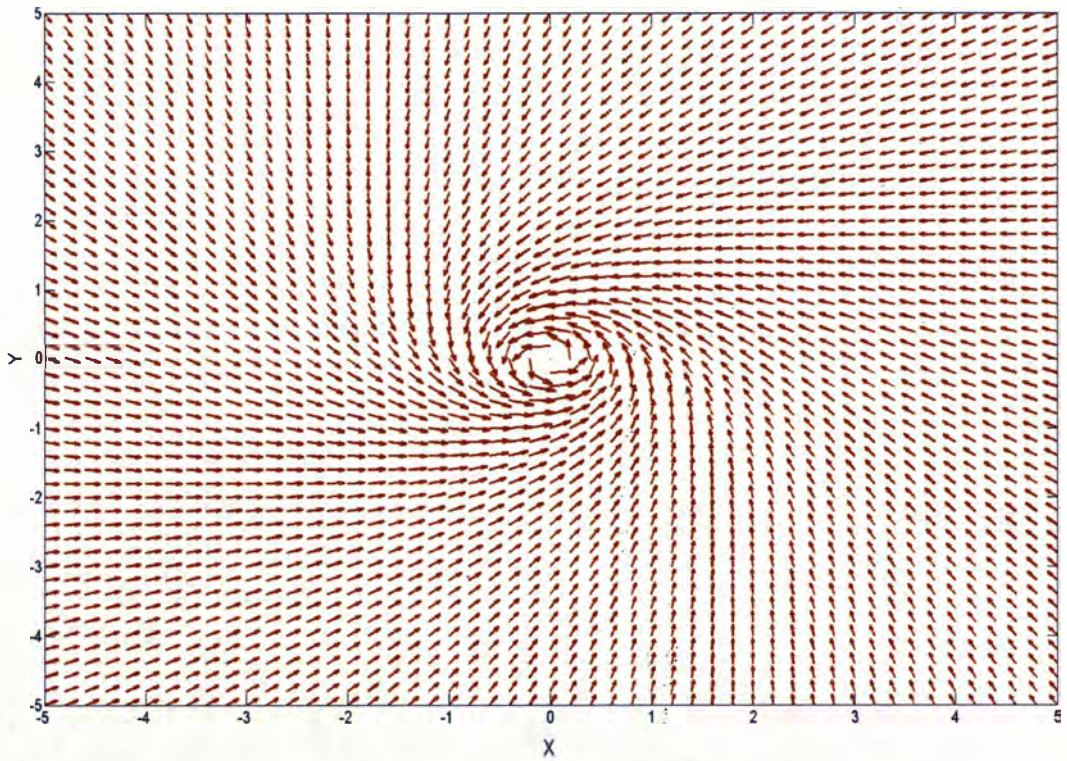


Figure 4.21: Velocity field showing the direction of $\vec{\alpha}$ when agent is turning counter-clockwise

A pedestrian's state changes from 'Exploring' to 'Moving to Target' (Section 4.3.3.4), since an opening has been identified for its next move and the pedestrian is moving towards it.

4.3.3.4 Moving to Target

In this state, the agent is moving towards a specific identified target. The target can be a shop entrance and an exit line segment, with the target point, a point belonging to those segments, or a pedestrian might be moving towards an opening that was selected from the previous section. When a pedestrian enters this state by having identified a shop to visit or he is moving towards an exit the agent's direction of movement is direct towards the selected point, thus minimising his energy expenditure. When a pedestrian reaches the shop he was aiming to visit his state changes to 'Entering Shop' (Section 4.3.3.5), while when he reaches an exit, his state becomes 'Exiting Environment' (Section 4.3.3.6). However when the pedestrian is moving towards an opening his state changes to 'Searching' (Section 4.3.3.2) in order to identify any possible shops that will satisfy his agenda. When a pedestrian has selected an opening that leads to a dead end and with no

shops satisfying its needs, it will continue moving till reaches the boundary of the environment at which it will turn 180 degrees and start searching again.

4.3.3.5 Entering Shop

An agent's state changes to entering a shop s when it was selected as a target and the agent reaches the shops entrance. At that point the type of shop is removed from the itinerary I_i of the agent. Then, the state of the agent changes to 'Searching'(Section 4.3.3.2). There is no waiting time inside a shop since we are interested in generating the footfall of the various street segments in the environment and we want the agents to be in constant movement. As such when a pedestrian touches a shop continues its journey to its next target.

4.3.3.6 Exiting Environment

Once an agent has an empty shopping agenda or its shopping time is over and reaches an exit, it is removed from the simulation environment.

4.4 Results

4.4.1 Implementation

The simulation was built in Java using the Geographical Information System OpenJump library [99]. The OpenJump library is an open-source project that provides controls which can be used to manipulate and display geographic information. In Figure 4.22 a screenshot of the application is being displayed, showing the GIS shape file of Kingston upon Thames centre (green line segments), as well as the entrances to the simulation environment (short bright green line segments), the exits (short red line segments) and the pedestrians (green dots).

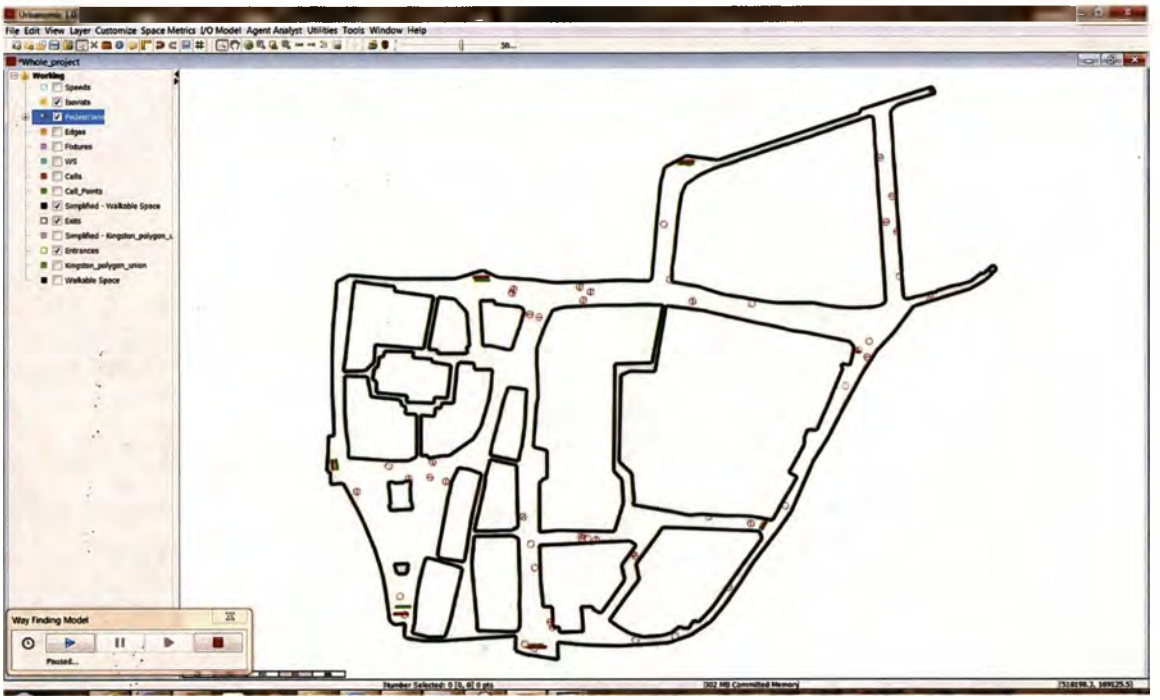
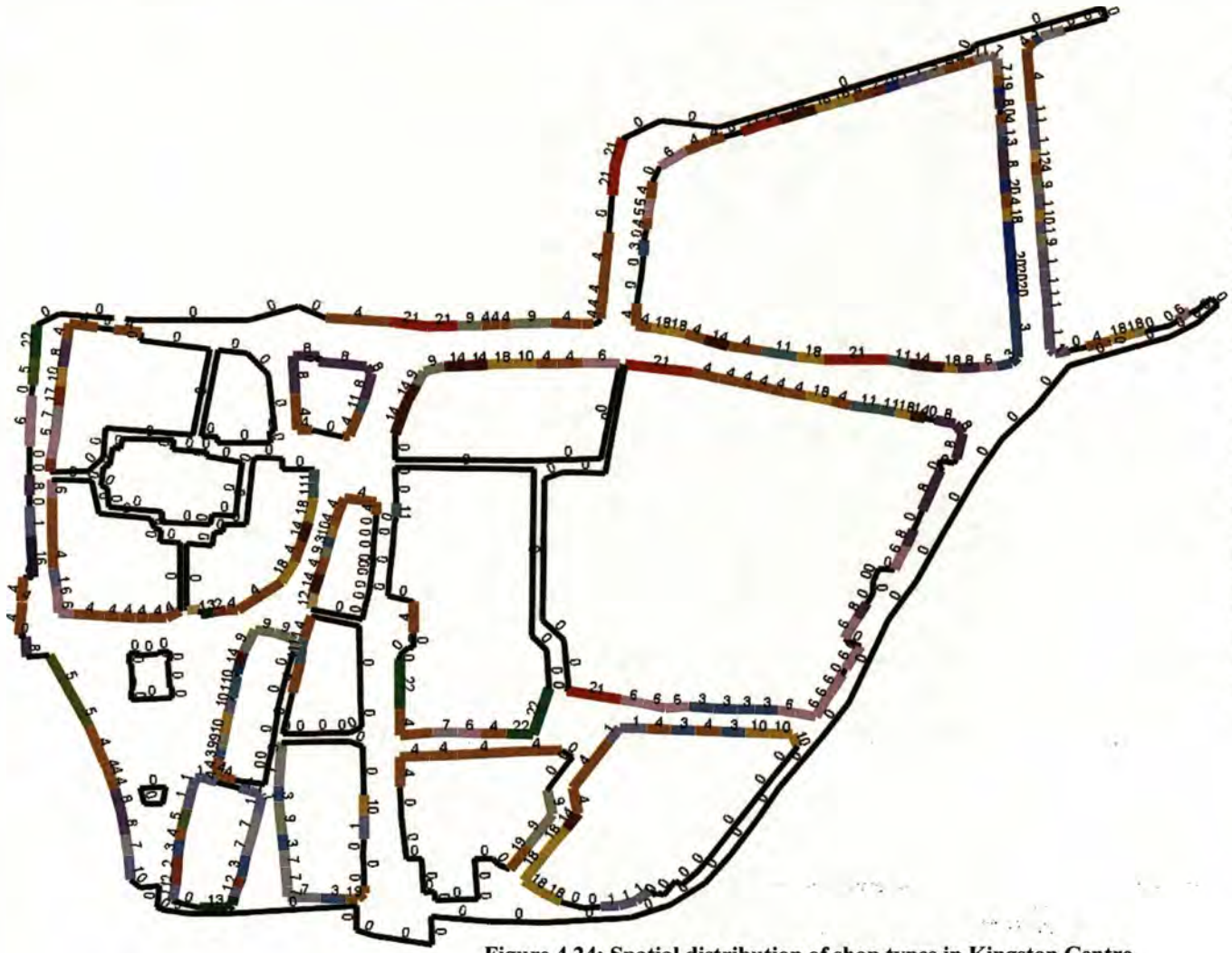


Figure 4.22: Application

4.4.2 Ground Truth

The ground truth for evaluating the methodology is obtained from the VOA business rates database of the valuation office agency website [134]. This contains the business rateable values of 2010 for properties in England and Wales. One of the major factors that affect the business rates is the footfall [47], which is the amount of people passing in front of a business premise, defines the popularity of a centre, and it is an indicator of potential spending. Thus using the rateable values we can infer the footfall of the various shops. In Figure 4.23, the rent values of Kingston upon Thames are displayed in a heat map with bright red values indicating higher rent values, while bright yellow indicate lower values. The total area of the simulated walkable space is 28446 square metres with 325 shops being present. There have been identified 22 different types of shops and the type distribution can be seen in Figure 4.24.

Figure 4.23: Kingston centre rent values. Light yellow indicates lower while dark red higher rateable values.



Description of Shop	Shop Type
No Shop	0
Fast Food	1
Cosmetic Services	2
Food Shop	3
Clothes Shop	4
Hardware	5
Furniture	6
Restaurant	7
Bank	8
Jewellery	9
Stationary	10
Pharmacy	11
Travel	12
Optical	13
Shoe Shop	14
Locksmith	15
Toy Store	16
Law Firm	17
Electronics	18
Medical Services	19
Betting	20
Multi-store 1	21
Multi-store 2	22

Figure 4.24: Spatial distribution of shop types in Kingston Centre.

4.4.3 *Evaluation Metric*

In order to evaluate the simulation, the footfall for each shop in the environment needs to be estimated. This is achieved by fitting a grid to the walkable space of the environment with each individual cell occupying one square metre surface. Then, a histogram is created for all cells, and a vote is cast to the bin of a cell if at a step of the simulation run a pedestrian was present at the particular cell. The votes of each bin are then divided by the total steps of the complete simulation run, and each cell is assigned a normalised occupancy value representing the proportion of time that it was occupied. For each line segment λ_k corresponding to a shop a rectangular mask is applied in front of it, occupying floor of the walkable space, with maximum depth of ten metres and length the length of the λ_k. Finally in order to calculate the footfall of a shop the occupancies of all cells present in the mask are added and the result is divided by the length of λ_k. The result represents the average number of pedestrians present in front of the shop per metre of the shop's active facade. In places where the constructed mask is too big for the available space, thus occupying space which is not walkable, only the cells that are in the walkable space are used for the footfall calculation. The final division over the length of a line segment is performed in order to remove the bias assigning higher footfall to longer line segments.

In order to compare the footfall of each shop against the normalised rent values, we use the Pearson product-moment correlation coefficient r which is calculated as such:

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})\cdot(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\cdot\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \quad (4.13)$$

4.4.4 *Experiments*

For the experiments we used two different types of behavioural pattern, agents with and agents without an itinerary. The agents with itinerary are obliged to search and visit shops of a specific type, based on their agenda, while the agents without an itinerary are wondering around the environment for a given time. We used these two different behavioural patterns in order to investigate the possible existence of a bias towards increased footfall on the streets with popular types of shops (i.e. Multi-Store 1 and 2 act as

wildcards for one destination of the pedestrians). Furthermore we are interested in investigating whether the spatial arrangement of the buildings alone and how this is perceived by the agents is sufficiently influential in determining the spatial distribution of footfall. Based on observations, eight locations were identified as enter and exit points (Figure 4.22). These population control points signify portals to either train or bus stations, parking spaces, or popular pedestrianised pathways. Furthermore in order to generalise our model and make it independent of the positions of entry and exit points, we performed another series of experiments with overlapping the whole walkable space with a grid of one by one metre cells, where each cell is a possible entry or exit point. During the simulation runs, at any time there are 50 agents present, where each agent has an infinite length field of view and each simulation run lasts 180 minutes of simulation time. The field of view of each pedestrian is set to 150 degrees.

4.4.4.1 Agents agenda

The agenda of each pedestrian is created by drawing a variable number of samples (uniform distribution), for both the number and the types of shops. The minimum value for the length of an itinerary is 1 while the maximum amount of shops that a pedestrian can visit is 10. Furthermore there are 22 different types of shops that a pedestrian can visit. After a pedestrian visits a shop, the type of the shop visited is removed from its agenda. If there is more than one of the same type of shop in his agenda, only one entry is removed. As mentioned before a shopping mall as well as a multistore (shop types 21 and 22) in the environment act as wildcards that can satisfy any type of shop from the agent's agenda, therefore they tend to be popular, however they act as wildcards only for one destination per pedestrian. A pedestrian after visiting all types of shops in his itinerary then searches for an exit and upon exiting the environment another agent is created substituting the exiting one. In the absence of an agenda the agents are just wondering inside the walkable space in a continuous exploratory state. Each pedestrian's trip is limited to 20 minutes after which the pedestrian is either searching for an exit (in the case of presence of population control points) or is just removed from the environment (in the case were pedestrians are generated from cells).

4.4.5 Experimental Results

In order to evaluate the performance of our method, we compare our approach of the pedestrian simulation against the approaches defined in [133]. Based on that paper the isovist of a pedestrian is segmented into non-overlapping bins, each covering 10° degrees of visual information. Openings that are present in the same bin are grouped together and by knowing the distance of each opening from the pedestrian eight different strategies for selecting an opening are presented

Name	Description
TU01	Choose the opening that is the furthest from the pedestrian.
TU02	Select randomly an opening.
TU03	For each bin in the field of view, choose the furthest opening. Then choose one of these at random.
TU04	For each group of 3 bins in the field of view choose the furthest opening. Then choose one of these at random.
TU05	Like TU04 but using group of 5 bins.
TU06	For each bin in the field of view, choose the furthest opening. Then choose one of these weighted according to how far it is.
TU07	For each bin in the field of view, choose the furthest opening. Then choose one of these weighted according to its angular deviation from the current course.
TU08	For each bin in the field of view, choose the furthest opening. Then choose one of these weighted according to its distance multiplied by the angular deviation from the current course.

Table 4.1: Opening selection approaches based in [133]

For the proposed approach, five different combinations of w_d and w_a (Equation 4.9) were used (see Figure 4.26) in order to assess which factor is more important. In Table 4.2-4.3 the correlation scores of our approach and the one in [133] are presented respectively, where on the left most column the presence of entrances (En: T) or cells (En :F) and agents with agenda (Ag: T) or without (Ag: F) is noted.

	$w_d = 1,$ $w_a = 0.1$	$w_d = 1,$ $w_a = 0.5$	$w_d = 1,$ $w_a = 1$	$w_d = 0.5,$ $w_a = 1$	$w_d = 0.1,$ $w_a = 1$
En: F , Ag: F	0.372305	0.459264	**0.598423**	0.503543	0.442106
En: F , Ag: T	0.351109	0.472863	**0.642525**	0.529684	0.469634
En: T , Ag: F	0.401987	0.463412	**0.602994**	0.496794	0.482396
En: T , Ag: T	0.390643	0.443482	**0.648694**	0.549693	0.420902

Table 4.2: Correlation scores for our method. The highest score per environmental configuration is indicated with bold letters.

	TU01	TU02	TU03	TU04	TU05	TU06	TU07	TU08
En: F , Ag: F	0.068098	0.184568	0.619999	0.400393	0.390965	**0.674906**	-0.02948	-0.17882
En: F , Ag: T	0.186848	0.213906	0.546538	0.534517	0.288278	**0.708963**	-0.16631	-0.16665
En: T , Ag: F	0.100832	0.177247	0.637	0.561038	0.446914	**0.646454**	-0.27258	-0.15338
En: T , Ag: T	0.100612	0.133404	0.644416	0.524315	0.407525	**0.682975**	-0.07463	-0.20203

Table 4.3: Correlation scores for methods of [133]. The highest score per environmental configuration is indicated with bold letters.

As it can be seen from the experimental results, TU06 performs the best with an average correlation of 0.708963 while our approach achieves an average correlation of 0.648694 performing slightly better that TU03 which has an average correlation of 0.644416. TU06 and TU03 are both favouring distant openings. Using TU03 distant opening of different bins have equal probability to be selected while TU06 favours probabilistically the openings who lie further. Thus long streets with many access points to them are being favoured by using these two approaches, which in our case the experimental environment as it can be seen in Figure 4.23 falls into this category. In Figure 4.25 the footfall generated by using TU06 is displayed.

Figure 4.25: Footfall generated by using TU06 method (En:T , Ag:T)

By observing the results from the experiments using our methodology we can see that the angular characteristic of an opening is more important than its distance characteristic.

Although the best results are obtained when they have equal weight in the decision process of a pedestrian, reducing w_a seems to affect more steeply the footfall generated from the simulation. Figure 4.26 shows the footfall generated by using our method.

Figure 4.26: Footfall generated by using our method (En: F , Ag: T, $w_d = 1$, $w_a = 1$)

From the results we can also observe that there is no clear indication if the method with which the pedestrians enter and exit the environment has an effect in the correlation between the observed footfall and the ground truth. Comparing experiments where only the entrance method changed (i.e. En:F or En:T), 15 of the experiments produced a higher correlation with the ground truth when pedestrians were entering the environment from the designated eight entrances while at 11 of the experiments the cell method performed better. The presence of the entrances, as these appear in Figure 4.22, encloses the simulation environment and the walkable space, and along with the fact that the entry distribution of pedestrians is uniform across the various entrances has as result to create an initial almost

uniform distribution of pedestrians in the periphery of the walkable space. Thus, although the entrances identify indeed the main entrance and exit points of the simulation environment, estimation of their true impact is infeasible due to lack of real origin/destination data.

A similar observation can be made for the presence or absence of agenda. Out of all experiments, when pedestrians had an agenda, 14 produced better correlation with the ground truth data while 12 performed better with no pedestrian agenda. Since the agendas of the pedestrians are created randomly sampled from a uniform distribution there and without any information regarding actual shopping preferences of pedestrians, the presence of an agenda does not add any real value in the simulation and the behaviour of the pedestrians with agenda is similar with the behaviour were they just explore the space.

Finally an interesting observation is the negative correlation that TU07 and TU08 scores. Although in [133] these methods achieve the highest score, in our experiments perform the worst. These approaches favour opening selections which have larger angular deviation from the current direction of the pedestrian, thus forcing the pedestrians to turn more often. Looking at the footfall produced in Figure 4.27 this behaviour can be observed as the pedestrians seem to be walking in loops.

Figure 4.27: footfall produced by using TU08 (En:T, Ag:F)

4.5 Conclusion

A novel methodology has been proposed in modelling and simulating pedestrian's shopping behaviour. A pedestrian, simulated by an intelligent agent, forms its decisions based on the spatial relationships of the various structures in the environment as these are perceived through its field of vision. In order to validate the model, the simulation parameters were adjusted using the business rates datasets, for the shops in the town centre of Kingston. Furthermore our route choice model was validated against a competitive one and although did not perform as accurately as it, it's correlation score indicates that indeed the arrangement of structures on the space and the geometrical relationships they form with each other has a effect in the pedestrian route choices. Turner's method favours movement in long axial lines and as we can observe from our environment the highest business rates

is along the street segments that offer the longest possible straight walk, however in environments where the highest footfall in not observed in the main axial lines, his method might fail. One direction of future work would be to incorporate the distance from the various targets in the environment as parameter to our model. Moreover a very interesting idea would be to use the computer vision methods presented in the previous chapters to auto calibrate the route choice model and the way pedestrians choose which shop to visit. The fusion of computer vision with pedestrian simulation is not something new, as approaches like [69] have already created basic simulation systems which adapt to measurements from camera observations.

CHAPTER FIVE

5. Conclusions and Future Work

The main goal of this thesis was to investigate new methods for data acquisition that can be used in combination with a simulation model, thus infusing to it real world measurements, for the creation of more close to reality simulation models. Moreover the creation of a simulation framework based on pedestrian visual perception in multiple attractor environments attempted to bring true human perception in the simulation model. In the following sections a summary of the problem tackled along with our proposed solution and contributions as well as any future work for each technical chapter presented in this thesis is given.

5.1 Pedestrian Speed Estimation

Automatic pedestrian speed estimation is a valuable tool for any pedestrian simulation framework; however the predictions of such a tool must have high accuracy in order for the simulation to be validated successfully against the reality. Using computer vision for the aforementioned task is a challenging task. Occlusions of people, different camera views, the presence of other class objects in an image, changes in the background environment and other issues can make the task even harder. Our proposed methodology, although does not address all the issues that a computer vision system built for speed estimation might face; it proposes a solution for speed estimation of unimpeded pedestrians and generates accurate estimations that can be used in a simulation model as speed profile distributions for the agents. In both systems implemented initially a tracker to each individual person is being applied to estimate his/her position in the environment, followed by a foot localisation algorithm. With the use of calibration data and by filtering undesirable pedestrian measurements it generates an accurate speed profile of the population observed.

As mentioned before the scope of this research was to measure the speed of single unimpeded pedestrians. However as simulations are getting enriched in data and information, the need to simulate pedestrian groups, as wells mobile impaired people, people with suitcases or generally carrying objects that restrict their speed, creates the need

for speed profiles for all these various classes. As such future work will be to be able to measure the speed profiles of all these intra-class variations.

5.2 People Counting

Similar to pedestrian speed estimation, people counting is a valuable tool for configuring a pedestrian simulation framework with realistic values. It can provide the information on how many people enter the simulation environment at any time from a given point, measure the distribution of pedestrians in the space, provide with information from which the simulation environment can infer the origin-destination of people and also infer the route choices of people. The task of people counting using computer vision suffers from the same issues as the pedestrian speed estimation. Furthermore since we are actually mostly interested in measuring the count of people in crowded environments, a counting system must be able to deal with these kinds of situations. There has been a lot research focusing in people counting, and lately with the high interest of the research community for the use of Convolutional Neural Networks, two solutions were proposed taking advantage the spatial invariance property of the CNNs and use their networks as human detectors. In our proposed methodology we argue that it is the environment itself that indicates where the presence of a person is, and as such we train the CNN on the whole image information and thus create relationships between the background and the foreground in the image. Furthermore, by combining the count information in the temporal domain we can further enhance the precision of our counting system.

The proposed solution needs further improvement in its accuracy and as such future work will be to enhance its precision. A way forward will be the fusion of image with temporal information, such as optical flow, to provide a better insight into the network regarding the association between frames of the presence of humans in the environment. Furthermore the examination of the applicability of networks with memory, such as the recurrent neural networks, can be used to learn a better representation of the background and thus make the foreground information easier to detect.

5.3 Modelling Pedestrian Shopping Behaviour

Modelling pedestrian shopping behaviour is used to predict the behaviour of people in multiple attractor environments where each individual has its own agenda of actions. Traditional approaches emphasize mainly in the set of activities that a pedestrian has to complete and do not take as much in consideration the true information perceived (i.e. what a person sees) by a pedestrian. Moreover the lack of real world data infused into the models, makes them vulnerable to the human interpretation of crowd dynamics and people behaviour in general. In this thesis we propose a framework for pedestrian shopping behaviour which fuses the visual perception of pedestrians used for route choices and target selection, with an underlying model of a multiple attractor environment where each pedestrian has a set of activities to complete. The route choice model we used displayed high correlation with the actual footfall distribution in the environment thus indicating that the topological arrangement of the buildings in the environment and their inter-relationship is a good indicator for predicting people's movement in the space.

The obvious way to move forward would be to fuse the three research topics described in this thesis to create a fully auto-calibrated model to simulate pedestrian behaviour. The counts of pedestrians for different locations can provide the data to infer the route choices of people and the places they visited, while speed estimation can be used to measure the speed of pedestrians under different conditions (e.g. if it is raining the speed of people is different from walking on a sunny day). Thus having this information can assist us to automate the calibration of our model to match the observed data.

5.4 Epilogue

This thesis aimed to examine ways to analyse pedestrian characteristics using computer vision and to investigate the correlation between pedestrian choices using visual perception in complex environments. For this reason novel frameworks were developed, each one aiming to a specific problem but with a motivation in the future to unite them for a better understanding of pedestrian behaviour.

Bibliography

[1] Albiol, Antonio, Maria Julia Silla, Alberto Albiol, and José Manuel Mossi. "Video analysis using corner motion statistics." In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 31-38. 2009.

[2] Antonini, Gianluca, Michel Bierlaire, and Mats Weber. "Discrete choice models of pedestrian walking behavior." *Transportation Research Part B: Methodological* 40, no. 8 (2006): 667-687.

[3] Antonini, Gianluca, Santiago Venegas Martinez, Michel Bierlaire, and Jean Philippe Thiran. "Behavioral priors for detection and tracking of pedestrians in video sequences." *International Journal of Computer Vision* 69, no. 2 (2006): 159-180.

[4] Applied Information Group. Legible London: A wayfinding study, London, 2006.

[5] Barnich, Olivier, and Marc Van Droogenbroeck. "ViBe: A universal background subtraction algorithm for video sequences." *Image Processing, IEEE Transactions on* 20, no. 6 (2011): 1709-1724.

[6] Bastien, Frédéric, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. "Theano: new features and speed improvements." *arXiv preprint arXiv:1211.5590* (2012).

[7] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." In *Computer Vision–ECCV 2006*, pp. 404-417. Springer Berlin Heidelberg, 2006.

[8] Ben-Akiva, Moshe, and Michel Bierlaire. "Discrete choice methods and their applications to short term travel decisions." In *Handbook of transportation science*, pp. 5-33. Springer US, 1999.

[9] Benedikt, Michael L. "To take hold of space: isovists and isovist fields."*Environment and Planning B* 6, no. 1 (1979): 47-65.

[10] Benezeth, Yannick, Pierre-Marc Jodoin, Bruno Emile, Hélène Laurent, and Christophe Rosenberger. "Comparative study of background subtraction algorithms." *Journal of Electronic Imaging* 19, no. 3 (2010): 033003-033003

[11] Bergstra, James, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. "Theano: a CPU and GPU math expression compiler." In *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4, p. 3. 2010.

[12] Berrou, J.L., Beecham, J., Quaglia, P., Kagarlis, M.A., and Gerodimos, A. Calibration and validation of the Legion simulation model using empirical data, PED 2005 Conference in Vienna Austria. 2005.

[13] Bertram, John EA. "Constrained optimization in human walking: cost minimization and gait plasticity." *Journal of experimental biology* 208, no. 6 (2005): 979-991.

[14] Bianco, Simone, Gianluigi Ciocca, and Raimondo Schettini. "How Far Can You Get By Combining Change Detection Algorithms?." *arXiv preprint arXiv:1505.02921* (2015).

[15] Bierlaire, Michel, Gianluca Antonini, and Mats Weber. "Behavioral dynamics for pedestrians." In *Moving Through Nets: The Physical and Social Dimensions of Travel. 10th International Conference on Travel Behavior Research. Lucerne.* 2003.

[16] Borgers, Aloys WJ, I. M. E. Smeets, A. D. A. M. Kemperman, and H. J. P. Timmermans. "Simulation of micro pedestrian behaviour in shopping streets."*van Leeuwen JPH, Timmermans J P. Progress in Design & Decision Support Systems. Heeze, The Netherlands* (2006): 101-116.

[17] Bouchrika, I., Nixon, M. S. People detection and recognition using gait for automated visual surveillance, in IET Conf. on Crime and Security, pp. 576-581, 2006.

[18] Bourdev, Lubomir, Subhransu Maji, Thomas Brox, and Jitendra Malik. "Detecting people using mutually consistent poselet activations." *Computer Vision–ECCV 2010* (2010): 168-181.

[19] Bradski, G., Kaehler A. . Learning OpenCV: Computer Vision with the OpenCV library. O'Reilly Media. 2008.

[20] Breiman, Leo. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.

[21] Bresenham, Jack E. "Algorithm for computer control of a digital plotter." *IBM Systems journal* 4, no. 1 (1965): 25-30.

[22] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 452–466, Apr. 2002.

[23] Chan, Antoni B., and Nuno Vasconcelos. "Modeling, clustering, and segmenting video with mixtures of dynamic textures." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30, no. 5 (2008): 909-926.

[24] Chan, Antoni B., Z-SJ Liang, and Nuno Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking." In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1-7. IEEE, 2008.

[25] Chen, Ke, Chen Change Loy, Shaogang Gong, Tony Xiang, and Queen Mary. "Feature Mining for Localised Crowd Counting." In *BMVC*, vol. 1, no. 2, p. 3. 2012.

[26] Chen, Ke, Shaogang Gong, Tao Xiang, and Chen Loy. "Cumulative attribute space for age and crowd density estimation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2467-2474. 2013.

[27] Cho, Sang-Hyun, and Hang-Bong Kang. "Integrated multiple behavior models for abnormal crowd behavior detection." In *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, pp. 113-116. IEEE, 2012.

[28] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, IEEE International Conference on Pattern Recognition, vlo. 2, pp. 142-149, 13-15 June, 2000.

[29] Comaniciu, Dorin, and Peter Meer. "Distribution free decomposition of multivariate data." *Pattern Analysis & Applications* 2, no. 1 (1999): 22-30.

[30] Conte, Donatello, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento. "A method for counting people in crowded scenes." In*Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 225-232. IEEE, 2010.

[31] Conti, Francesco, Antonio Pullini, and Luca Benini. "Brain-inspired classroom occupancy monitoring on a low-power mobile platform." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pp. 624-629. IEEE, 2014.

[32] Cristani, Marco, Loris Bazzani, Giulia Paggetti, Andrea Fossati, A. D. Bue, D. Tosato, Gloria Menegaz, and Vittorio Murino. "Social interaction discovery by

statistical analysis of f-formations." In *Proceedings of British Machine Vision Conference*. 2011.

[33] Dalal, N., Triggs, B., . Histograms of Oriented Gradients for Human Detection, CVPR 2005, IEEE Computer Society Conference on Computer Vision and Pattern Recognition,1, pp. 886-893. 2005.

[34] Daamen, Winnie, Serge P. Hoogendoorn, and Piet HL Bovy. "First-order pedestrian traffic flow theory." *Transportation Research Record: Journal of the Transportation Research Board* 1934, no. 1 (2005): 43-52.

[35] Denman, Simon, Vinod Chandran, and Sridha Sridharan. "An adaptive optical flow technique for person tracking systems." *Pattern recognition letters* 28, no. 10 (2007): 1232-1239.

[36] J. Dijkstra, J. Jessurun, H. Timmermans, and B. de Vries. "A framework for processing agent based pedestrian activity simulations in shopping environments". In: 20th European Meeting on Cybernetics and Systems Research. Ed. by R. Trappl. Austrian Society for Cybernetic Studies, Vienna. , pp. 515–521. 2010.

[37] Dijkstra, Jan, Joran Jessurun, Harry Timmermans, and Bauke de Vries. "A Framework for Processing Agent-Based Pedestrian Activity Simulations in Shopping Environments." *Cybernetics and Systems* 42, no. 7 (2011): 526-545.

[38] Dollar, Piotr, Christian Wojek, Bernt Schiele and Pietro Perona. "Pedestrian detection: An evaluation of the state of the art." *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34, no. 4 (2012): 743-761.*

[39] Dollár, Piotr, Zhuowen Tu, Pietro Perona, and Serge Belongie. "Integral Channel Features." In *BMVC*, vol. 2, no. 4, p. 5. 2009.

[40] *Enzweiler, Markus, and Dariu M. Gavrila. "Monocular pedestrian* detection: Survey and experiments." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, no. 12 (2009): 2179-2195.

[41] Evans, Daphne, and Paul Norman. "Understanding pedestrians' road crossing decisions: an application of the theory of planned behaviour." *Health Education Research* 13, no. 4 (1998): 481-489.

[42] Felzenszwalb, Pedro F., Ross B. Girshick, David McAllester, and Deva Ramanan. "Object detection with discriminatively trained part-based models."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, no. 9 (2010): 1627-1645.

[43] Fiaschi, Luca, Ullrich Koethe, Rahul Nair, and Fred A. Hamprecht. "Learning to count with regression forest and structured labels." In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 2685-2688. IEEE, 2012.

[44] Foggia, Pasquale, Gennaro Percannella, Carlo Sansone, and Mario Vento. "A graph-based algorithm for cluster detection." *International Journal of Pattern Recognition and Artificial Intelligence* 22, no. 05 (2008): 843-860.

[45] Foltête, J.-C. and Piombini, A. . Urban Layout, Landscape Features and Pedestrian Usage.Landscape and Urban Planning 81(3): 225-234. 2007.

[46] Gaud, Nicolas, Stéphane Galland, Franck Gechter, Vincent Hilaire, and Abderrafiâa Koukam. "Holonic multilevel simulation of complex systems: Application to real-time pedestrians simulation in virtual urban environment."*Simulation Modelling Practice and Theory* 16, no. 10 (2008): 1659-1676.

[47] Genecon LLP and Partners, Understanding Hight Street Performance, Department for Business, Innovation and Skills, December 2011.

[48] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jagannath Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 580-587. IEEE, 2014.

[49] Goodfellow, Ian J., David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio. "Pylearn2: a machine learning research library." *arXiv preprint arXiv:1308.4214* (2013).

[50] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, changedetection.net: A new change detection benchmark dataset, in Proc. IEEE Workshop on Change Detection (CDW-2012) at CVPR-2012, Providence, RI, 16-21 Jun., 2012.

[51] Granié, Marie-Axelle. "Effects of gender, sex-stereotype conformity, age and internalization on risk-taking among adolescent pedestrians." *Safety science*47, no. 9 (2009): 1277-1283.

[52] Granié, Marie-Axelle, Marjorie Pannetier, and Ludivine Guého. "Developing a self-reporting method to measure pedestrian behaviors at all ages." *Accident Analysis & Prevention* (2012).

[53] Guy, Stephen J., Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming Lin, and Dinesh Manocha. "PLEdestrians: a least-effort approach to crowd simulation." In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 119-128. Eurographics Association, 2010.

[54] Haitovsky, Yoel. "On multivariate ridge regression." *Biometrika* 74, no. 3 (1987): 563-570.

[55] Handy, S. . Methodologies for exploring the link between urban form and travel behaviour. Transp. Res. Part D 1 (2), 151–165. 1996.

[56] Harris, Chris, and Mike Stephens. "A combined corner and edge detector." In*Alvey vision conference*, vol. 15, p. 50. 1988.

[57] Hashemzadeh, Mahdi, Gang Pan, and Min Yao. "Counting moving people in crowds using motion statistics of feature-points." *Multimedia Tools and Applications* (2013): 1-35.

[58] Heïgeas, Laure, Annie Luciani, Joelle Thollot, and Nicolas Castagné. "A physically-based particle model of emergent crowd behaviors." *arXiv preprint arXiv:1005.4405* (2010).

[59] Helbing, Dirk, Illes J. Farkas, Peter Molnar, and Tamás Vicsek. "Simulation of pedestrian crowds in normal and evacuation situations." *Pedestrian and evacuation dynamics* 21 (2002): 21-58.

[60] Helbing, Dirk, and Peter Molnar. "Social force model for pedestrian dynamics."*Physical review E* 51, no. 5 (1995): 4282.

[61] Hermant, L. F. L. "Human movement behaviour in South African railway stations: implications for design." *SATC 2011* (2011).

[62] Hou, Ya-Li, and Grantham KH Pang. "People counting and human detection in a challenging situation." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 41, no. 1 (2011): 24-33.

[63] S.P. Hoogendoorn, W. Daamen, P.H.L. Bovy, Extracting Microscopic Pedestrian Characteristics from Video Data, TRB2003 Annual Meeting. 2003.

[64] Huang, Ling, S. C. Wong, Mengping Zhang, Chi-Wang Shu, and William HK Lam. "Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm." *Transportation Research Part B: Methodological* 43, no. 1 (2009): 127-141.

[65] [Inui03] K. Inui, S. Kaneko, and S. Igarashi, "Robust line fi tting using LmedS clustering," *Systems and Computers in Japan* 34 (2003): 92–100

[66] Isard, M., Blake ,A. . Condensation – conditional density propagation for visual tracking, International Journal of Computer Vision, 29(1), pp. 5-28. 1998

[67] Ismail K., Sayed, T., Saunier, N. . Automated collection of pedestrian data using computer vision techniques. In Transportation Research Board Annual Meeting Compendium of Papers, January 2009.

[68] Johansson, Anders, Dirk Helbing, and K. SHUKLA PRADYUMN. "Specification of the social force pedestrian model by evolutionary adjustment to video tracking data." *Advances in Complex Systems* 10, no. supp02 (2007): 271-288.

[69] Junker, O., V. Strauss, R. Majer, and N. Link. "Real-time video analysis of pedestrians to support agent simulation of people behavior." In *Pedestrian and Evacuation Dynamics*, pp. 81-94. Springer US, 2011.

[70] KaewTraKulPong, P., Bowden, R., . An improved adaptive background mixture model for realtime tracking with shadow detection. In Proc. 2nd European Workshop on Advanced Video Bases Surveillance Systems, AVBS01. 2001.

[71] Kalman R. .A new approach to linear filtering and prediction problem, Journal of Basic Engineering, 82(1), pp. 35-45. 1960.

[72] Karamouzas, Ioannis, and Mark Overmars. "Simulating and Evaluating the Local Behavior of Small Pedestrian Groups." *IEEE Transactions on Visualization and Computer Graphics* (2011): 394-406.

[73] J. Kerridge, N. McNair (1999). PEDFLOW - A system for modelling pedestrian movement using occam. In BM Cook (ed), Architectures, Languages and Techniques for Concurrent Systems pp 1 -17. : IOS Press Amsterdam.

[74] Kitazawa, K., Batty, M. Pedestrian Behaviour Modelling. An Application to Retail Movements using a Genetic Algorithm. Centre for Advanced Spatial Analysis, University College London, 2004.

[75] Kong, Dan, Douglas Gray, and Hai Tao. "A viewpoint invariant approach for crowd counting." In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 1187-1190. IEEE, 2006.

[76] Kumagai, Shohei, and Kazuhiro Hotta. "HLAC between Cells of HOG Feature for Crowd Counting." In *Advances in Visual Computing*, pp. 688-697. Springer International Publishing, 2014.

[77] Lakoba, Taras I., David J. Kaup, and Neal M. Finkelstein. "Modifications of the Helbing-Molnar-Farkas-Vicsek social force model for pedestrian evolution."*Simulation* 81, no. 5 (2005): 339-352.

[78] Lam, W. H. K. , Cheung, C. .Pedestrian Speed/Flow Relationships for Walking Facilities in Hong Kong, Journal of Transportation Engineering, 126 (4), pp. 343-349. 2000.

[79] Leal-Taixe, Laura, Gerard Pons-Moll, and Bodo Rosenhahn. "Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker." In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 120-127. IEEE, 2011.

[80] LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.

[81] LeCun, Yann A., Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. "Efficient backprop." In *Neural networks: Tricks of the trade*, pp. 9-48. Springer Berlin Heidelberg, 2012.

[82] Legion Ltd, 2014, Legion #1 in Pedestrian Simulation Solutions, [online] Available at: http://www.legion.com. [Accessed at 10th November 2014]

[83] Lempitsky, Victor, and Andrew Zisserman. "Learning to count objects in images." (2010).

[84] Lipton, A. J., Fujiyoshi, H., Patil, R. S. . Moving target classification and tracking from real-time video, in Proc. IEEE Workshop Application of Computer Vision, 1998, pp. 8-14. 1998.

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

[86] Loy, Chen Change, Shaogang Gong, and Tao Xiang. "From semi-supervised to transfer counting of crowds." In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 2256-2263. IEEE, 2013.

[87] Maddalena, Lucia, and Alfredo Petrosino. "The SOBS algorithm: what are the limits?." In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 21-26. IEEE, 2012.

[88] Martinez-del-Rincon, J., Nebel, J.-C., Makris, D., Orrite, C.: Tracking Human Body Parts Using Particle Filters Constrained by Human Biomechanics. In: BMVC 2008 (2008).

[89] Masoud O and Papanikolopoulos N, A novel method for tracking and counting pedestrians in real-time using a single camera, IEEE Transactions on Vehicular Technology 50(5) (2001) 1267-1278.

[90] Mehran, Ramin, Alexis Oyama, and Mubarak Shah. "Abnormal crowd behavior detection using social force model." In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 935-942. IEEE, 2009.

[91] Meyer, D., Denzler, J., Niemann, H. . Model based extraction of articulated objects in image sequences for gait analysis, in Proc. IEEE Int. Conf. Image Processing, pp.78-81. 1998.

[92] Mohler, Betty J., William B. Thompson, Sarah H. Creem-Regehr, Herbert L. Pick Jr, and William H. Warren Jr. "Visual flow influences gait transition speed and preferred walking speed." *Experimental Brain Research* 181, no. 2 (2007): 221-228.

[93] Moon, Todd K. "The expectation-maximization algorithm." Signal processing magazine, IEEE 13, no. 6 (1996): 47-60.

[94] Moosmann, Frank, Bill Triggs, and Frederic Jurie. "Fast discriminative visual codebooks using randomized clustering forests." *Advances in Neural Information Processing Systems 19* (2007): 985-992.

[95] Musse, Soraia Raupp, and Daniel Thalmann. "Hierarchical model for real time simulation of virtual human crowds." *Visualization and Computer Graphics, IEEE Transactions on* 7, no. 2 (2001): 152-164.

[96] Nielsen, Michael. "Neural Networks and Deep Learning." *Neural Networks and Deep Learning. Determination Press* 1 (2014).

[97] K. Nummiaro, E. Koller-Meier, and L. Van Gool, A color based particle filter, in First International Workshop on Generative-Model-Based Vision, A.E.C. Pece, Ed., pp. 53-60. 2002.

[98] O'Connor, Shawn M., and J. Maxwell Donelan. "Fast visual prediction and slow optimization of preferred walking speed." *Journal of Neurophysiology* 107, no. 9 (2012): 2549-2559.

[99] OpenJump, accessed October 19, 2013, http://www.openjump.org

[100] Papadimitriou, Eleonora, George Yannis, and John Golias. "A critical assessment of pedestrian behaviour models." *Transportation Research Part F: Traffic Psychology and Behaviour* 12, no. 3 (2009): 242-255.

[101] Paris, Sébastien, and Stéphane Donikian. "Activity-driven populace: a cognitive approach to crowd simulation." *Computer Graphics and Applications, IEEE* 29, no. 4 (2009): 34-43.

[102] PathIntelligence, 2011, Pedestrian path measurement technology, [online] Available at: http://www.pathintelligence.com. [Accessed at 10th November 2011]

[103] Patzold, Michael, Rubén Heras Evangelio, and Thomas Sikora. "Counting people in crowded environments by fusion of shape and motion information." In*Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 157-164. IEEE, 2010.

[104] Pelechano, Nuria, Jan M. Allbeck, and Norman I. Badler. "Virtual crowds: Methods, simulation, and control." *Synthesis Lectures on Computer Graphics and Animation* 3, no. 1 (2008): 1-176.

[105] Pellegrini, Stefano, Andreas Ess, Konrad Schindler, and Luc Van Gool. "You'll never walk alone: Modeling social behavior for multi-target tracking." In*Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261-268. IEEE, 2009.

[106] Pellegrini, Stefano, Andreas Ess, and Luc Gool. "Predicting pedestrian trajectories." *Visual Analysis of Humans* (2011): 473-491.

[107] Perbet, F., Maki, A., Stenger, B, . Correlated probabilistic trajectories for pedestrian motion detection, 12th IEEE International Conference on Computer Vision. 2009.

[108] Perko, Roland, Thomas Schnabel, Gerald Fritz, Alexander Almer, and Lucas Paletta. "Counting people from above: Airborne video based crowd analysis."*arXiv preprint arXiv:1304.6213* (2013).

[109] Pham, Viet-Quoc, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. "COUNT Forest: CO-Voting Uncertain Number of Targets Using Random Forest for Crowd Density Estimation." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3253-3261. 2015

[110] Polana, R., Nelson, R. . Low level recognition of human motion, in Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, 1994, pp. 77-82. 1994.

[111] Queen Mary University of London. "Mall Dataset". http://www.eecs.qmul.ac.uk/ http://www.eecs.qmul.ac.uk/~ccloy/downloads_mall_dataset.html (accessed September 1, 2015).

[112] Radke, Richard J., Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. "Image change detection algorithms: a systematic survey." *Image Processing, IEEE Transactions on* 14, no. 3 (2005): 294-307.

[113] Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

[114] Rosenbloom, Tova. "Crossing at a red light: Behaviour of individuals and groups." *Transportation Research Part F: Traffic Psychology and Behaviour* 12, no. 5 (2009): 389-394.

[115] Russell, Stuart. "Artificial Intelligence: A Modern Approach Author: Stuart Russell, Peter Norvig, Publisher: Prentice Hall Pa." (2009).

[116] Ryan, David, Simon Denman, Clinton Fookes, and Sridha Sridharan. "Crowd counting using multiple local features." In *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.*, pp. 81-88. IEEE, 2009.

[117] Scovanner, Paul, and Marshall F. Tappen. "Learning pedestrian dynamics from the real world." In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 381-388. IEEE, 2009.

[118] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *Proc. of IEEE International Conference on Video and Signal Based Surveillance*, 2006.

[119] Shi, J., Tomasi, C. . Good features to track, 9th IEEE Conference on Computer Vision and Pattern Recognition, pp. 593 – 600. June 1994.

[120] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowdes situations. In *Proc. of IEEE International Conference on Video and Signal Based Surveillance*, 2006.

[121] An, Senjian, Wanquan Liu, and Svetha Venkatesh. "Face recognition using kernel ridge regression." In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1-7. IEEE, 2007.

[122] Shao, Wei, and Demetri Terzopoulos. "Autonomous pedestrians." *Graphical Models* 69, no. 5 (2007): 246-274.

[123] S Singh, S.A. Velastin and H Ragheb - "MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods" in 2nd Workshop on Activity monitoring by multi-camera surveillance systems (AMMCSS), August 29, Boston, USA, (2010).

[124] Sourtzinos, Panagiotis, Dimitrios Makris, and Paolo Remagnino. "Highly accurate estimation of pedestrian speed profiles from video sequences."*Innovations in Defence Support Systems–3* (2011): 71-81.

[125] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15, no. 1 (2014): 1929-1958.

[126] St-Charles, Pierre-Luc, Guillaume-Alexandre Bilodeau, and Robert Bergevin. "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity." *Image Processing, IEEE Transactions on* 24, no. 1 (2015): 359-373.

[127] Subburaman, Venkatesh Bala, Adrien Descamps, and Cyril Carincotte. "Counting people in the crowd using a generic head detector." In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pp. 470-475. IEEE, 2012.

[128] Sun, Quanbin, and Song Wu. "A Configurable Agent-Based Crowd Model with Generic Behavior Effect Representation Mechanism." *Computer-Aided Civil and Infrastructure Engineering* (2014).

[129] Sullman, M. J. M., M. Eugenia Gras, Silvia Font-Mayolas, L. Masferrer, Monica Cunill, and Monserrat Planes. "The pedestrian behaviour of Spanish adolescents." *Journal of adolescence* 34, no. 3 (2011): 531-539.

[130] Sutskever, Ilya, James Martens, George Dahl, and Geoffrey Hinton. "On the importance of initialization and momentum in deep learning." In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pp. 1139-1147. 2013.

[131] Tekalp, A. Murat, and A. Murat Tekalp. *Digital video processing*. Vol. 1. Upper Saddle river, NJ: Prentice Hall PTR, 1995.

[132] Tsai, R. Y. .A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, IEEE Journal of Robotics and Automation 3, pp. 323-344. 1987.

[133] Turner, Alasdair. "The ingredients of an exosomatic cognitive map: Isovists, agents and axial lines?." (2007): 163-180.

[134] Valuation Office Agency Business Rates, accessed Novemeber 1st, 2012 http://www.2010.voa.gov.uk/rli/

[135] VAUGHAN, C. L., B. L. DAVIS, and J. C. O'CONNOR. *Dynamics of Human Gait*. Champaign, IL: Human Kinetics, 1992, pp. 26.

[136] Willis, R. Kukla, J. Kerridge and J. Hine, Laying the foundations: The use of Video Footage to Explore Pedestrian Dynamics in PEDFLOW, PED2001 proceedings 181-186. 2001.

[137] Wu, B., Nevatia, R. .Detection and tracking of multiple partially occluded humans by bayesian combination of edgelet based part detectors, International Journal of Computer Vision, 75(2), pp. 247-266. 2007.

[138] Yamaguchi, Kota, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. "Who are you with and Where are you going?." In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1345-1352. IEEE, 2011.

[139] Yao, Jian, and J-M. Odobez. "Multi-layer background subtraction based on color and texture." In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1-8. IEEE, 2007.

[140] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey."*Acm Computing Surveys (CSUR)* 38, no. 4 (2006): 13.

[141] Wikipedia contributors, "Isovist," *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Isovist&oldid=544069901 (accessed October 11, 2013)

[142] Wilensky, Uri. "NetLogo. Evanston, IL." *Center for Connected Learning and Computer-Based Modeling, Northwestern University. ccl. northwestern. edu/netlogo* (1999).

[143] J. Zacharias, T. Bernhardt, et al., "Computer-Simulated Pedestrian Behavior in Shopping Environment," Journal of Urban Planning and Development **131(3)**, 195-200 (2005).

[144] Zhang, Cong, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. "Cross-scene crowd counting via deep convolutional neural networks." In *Proc. CVPR*. 2015.

[145] Zhang, Z., Wang, M. and Geng, X., 2015. Crowd counting in public video surveillance by label distribution learning. *Neurocomputing*, *166*, pp.151-163.

[146] Zhu, Wei, and Harry Timmermans. "Modeling pedestrian shopping behavior using principles of bounded rationality: model comparison and validation."*Journal of Geographical Systems* 13, no. 2 (2011): 101-126.