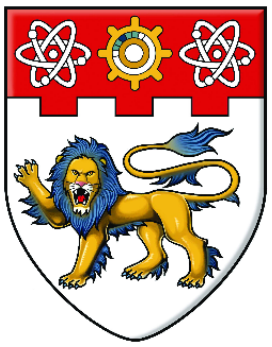


NANYANG
TECHNOLOGICAL
UNIVERSITY

Visual Detection and Crowd Density Modeling of Pedestrians

A thesis submitted
for a degree of Doctor of Philosophy
by

Tan Sing Kuang

School of Computer Engineering
Nanyang Technological University

October 20, 2017

Abstract

This thesis attempts to address two problems that are related to the sensing and prediction of pedestrian distributions in urban settings. The first research topic is on the automatic collection of pedestrian data, to augment the information available to urban planners. The second research topic is on automatically predicting the pedestrian density distributions given planned floor layouts of malls, potentially allowing architects to interactively adapt their designs and avoid excessively congested or underutilized regions.

In the first part of the thesis, we will address on the problem of detecting pedestrians in camera images. The challenges faced now are large variations of appearances and poses, differences in illumination, occlusions and cluttered background. We tackle this by introducing a novel feature that captures second order intensity variations, which can complement existing HOG (Histogram of Oriented Gradients) and LBP (Local Binary Patterns) features. This has shown improvements in detection accuracy over some frequently used datasets. Visualization of example detection responses due to different features and weights are provided to more intuitively explain the reasons behind the improved performance.

In the second part of the thesis, we model and predict the approximately steady-state pedestrian density distributions in buildings. These are affected by a large number of latent variables such as the popularity of different shops and different possible routes that shoppers may take between shops. We proposed a probabilistic model that establishes the Markovian relationship between the different latent variables and parameters. We validated the predictions against ground truth pedestrian counts, and also analyzed how the predicted popularity of shops compared against measured traffic at shop entrances.

Acknowledgments

Throughout my work, many people have assisted me in some way or another.

First of all, I would like to thank my supervisor Prof. Cham Tat Jen for his continuous support and guidance of my Ph.D. study and research. He has spent a lot of time through each step of the research and has given me his invaluable guidance to write this thesis. I also want to thank my second supervisor Prof. Wu Jianxin for his guidance and knowledge on pedestrian detection.

Meanwhile, I want to thank my colleagues at Future Cities Laboratory. Especially I want to thank members of Module IX Simulation Platform for their support and assistance. I want to thank Zeng Wei for proofreading my papers and give suggestions. I also want to thank students from NTU CeMNet lab for giving their assistance on my papers, report, thesis and presentation slides.

Last but not least, I want to thank my family for their constant supports and encouragement. Specially, I would like to thank my parents, who make me always feel warm and sweet with their ultimate care and love.

Contents

Abstract i
Acknowledgments ii
List of Figures vii
List of Tables xiv
List of Abbreviations xvi
List of Notation xvii

1 Introduction **1**
1.1 Background 1
1.2 Motivation 3
1.3 Research Objectives and Contributions 6
1.4 Thesis Organization 8

2 Literature Review **9**
2.1 Crowd Sensing 9
2.1.1 Manual Counting 9
2.1.2 Global Positioning System (GPS) and Travel Survey 10
2.1.3 3D Sensing 10
2.1.4 Image-Based Pedestrian Sensing 10
2.1.5 Image-based Crowd Analysis 19
2.2 Crowd Modeling 21
2.2.1 Pedestrian Simulation 22
2.2.2 Trip Modeling 23
2.2.3 Analysis of Pedestrian Perception and Cognition 25

2.2.4 Modeling of Crowd Disasters . . . 25
2.3 Human Factors in Urban Design . . . 26
2.3.1 Space Syntax and Influence on Crowds . . . 26
2.3.2 Evaluation of Urban Design . . . 27

3 Pedestrian Detection Using Second Order Information 29
3.1 Overview . . . 29
3.2 The Hessian-HOG-LBP Model . . . 32
3.2.1 Hessian and Curvature . . . 32
3.2.2 Hessian Features . . . 34
3.3 Adaptation of Hessian Weights with HOG and LBP . . . 37
3.3.1 Visualization of SVM Responses . . . 38
3.4 Analytical Comparison of HOG, LBP and Hessian Features . . . 39
3.4.1 When Hessian is Better than HOG . . . 40
3.4.2 When Hessian is Better than LBP . . . 41
3.4.3 Detection Support from Different Parts of the Hessian Weights . . . 42
3.5 Detection Experiments . . . 52
3.5.1 Description of Datasets . . . 53
3.5.2 Numerical Results . . . 53
3.6 Summary . . . 62

4 Pedestrian Density Distribution Model 63
4.1 Overview . . . 63
4.1.1 Quasi-Stationarity of Pedestrian Density Distribution . . . 65
4.1.2 Contributions . . . 67
4.2 Modeling Framework, Assumptions and Conjectures . . . 68
4.2.1 Categorization of Vessels . . . 70
4.3 Pedestrian Density Distribution Modeling . . . 71
4.3.1 Overview . . . 71
4.3.2 Node Probabilities Conditioned on Paths . . . 75
4.3.3 Path Choice Preference . . . 75
4.3.4 Conditional Portal Popularity . . . 82

4.3.5 Conditional Vessel Popularity . . . 83
4.3.6 Vessel Category Popularity . . . 85
4.4 Likelihood of Pedestrian Counts . . . 88
4.5 Learning . . . 88
4.6 Inference . . . 90
4.7 Evaluation Methodology . . . 91
4.7.1 Data Collection of Pedestrian Counts . . . 91
4.7.2 Baseline Prediction . . . 92
4.7.3 Log Likelihood Ratio . . . 93
4.7.4 Neyman-Pearson Test . . . 93
4.7.5 Fisher's Method for Meta Analysis . . . 96
4.7.6 Experimental Setup . . . 97
4.8 Experimental Results . . . 97
4.8.1 Leave-One-Out Cross Validation Results . . . 97
4.8.2 In-Test Optimization . . . 100
4.8.3 Quantitative Results . . . 102
4.8.4 Interpolation with Path Preference Priors . . . 104
4.8.5 Evaluation of Across-Mall Learning . . . 106
4.8.6 Interpretation of Learned Path Preference Parameters . . . 107
4.8.7 Comparison to Agent-Based Pedestrian Simulation . . . 110
4.9 Summary . . . 112

5 Learning Category Popularities **114**
5.1 Overview . . . 114
5.2 The Physics of Popularity . . . 114
5.3 Invariant Factors in Area Vessel Flow . . . 116
5.4 Categorization of Vessels based on Flow Densities . . . 117
5.5 Learning Flow Densities of Area Vessel Categories . . . 119
5.5.1 Prediction Results for Pedestrian Density Distribution . . . 120
5.5.2 Data Collection of Actual Customer Flows . . . 121
5.5.3 Comparison of Model-based Category Popularities to Measured Flows 125
5.5.4 Discussion . . . 132
5.6 Summary . . . 134

6 Conclusions and Future Work **135**

6.1 Conclusions 135

6.2 Future Work 137

6.2.1 Data Acquisition 137

6.2.2 Modeling 138

6.2.3 Automatic Optimization of Building Layouts 139

A Appendix **141**

A.1 Proof that Uniform Distribution is the Highest Expected Log Likelihood Value from Counts Sampled from Any Possible Multinomial Distribution 141

A.2 Interleaved Optimization Algorithm for Learning Path Preference Parameters $(\mathbf{a};\mathbf{b};\mathbf{c})$ and Non-Area Vessel Category Popularities $\mathbf{f}$ 142

A.3 Visual Prediction Results from Learned Vessel Category Popularities f_γ 144

A.4 Graphical Interpretation of Neyman Pearson Test 146

A.5 Theoretical Timing for Generating Steady State Pedestrian Density Distribution 147

A.6 Algorithm to Calculate the Total Likelihood Ratio of q_γ Across All the Training Floors 148

References **153**

Publications **173**

List of Figures

1.1 Detection results from our pedestrian detector. Purple boxes are the false positives. In this figure, a correct detection is recorded if the detected bounding box intersects the ground truth bounding box by more than 50% of the total combined area of the two bounding boxes, following established norms for experimental evaluation (see chapter 3). 3

1.2 Inferring crowd density distribution given a building layout. Differences in color hues mean different density distribution values as shown in the color bar. Top right is ground truth. Floor layout taken from ION Orchard app. 4

1.3 Thesis framework 6

2.1 An example of a zoning plan. This plan is for the Grand Gateway 66 shopping mall. Image taken from http://www.grandgateway66.com/. 28

3.1 Comparing the image intensity surfaces from the person (top) and non-person (bottom) images. Notice that the patches on both person and non-person images have the same HOG and LBP patterns, but they have different intensity curvatures. These curvatures are useful to discriminate between the two images. ©2015 IEEE 30

3.2 Example of an image seen as a function of two variables where the output of the function is the pixel intensity value (left) and the original image (right). . . 33

3.3 Types of curvature found in a small image patch. From top to bottom and left to right: local maximum, local minimum, saddle, ridge, valley and flat surface. 33

3.4 Block diagram of our detector. 36

3.5 3D intensity surface of an image. ©2015 IEEE 36

3.6 Left to right: First pair shows negative and positive weights for Hessian with 1st eigenvalue negative in value, second pair for 1st eigenvalue positive in value, third pair for 2nd eigenvalue negative in value, fourth pair for 2nd eigenvalue positive in value. (a): A detector trained on Hessian features only. (b): A detector trained on Hessian, HOG and LBP features. Some parts of Hessian is used to detect the shape of a human and some parts of Hessian detect curvature and offload the detection of human shape to HOG. ©2015 IEEE 38
3.7 Example of comparing the responses from individual components of the combined descriptor as compared to the Hessian-only descriptor detection. Redder colors indicate locations of the image for which the features respond positively while greener colors mean the opposite. The labels "HOG", "Hessian" and "LBP" refer to the different features used within the detector. The numbers below the HOG, Hessian and LBP feature outputs are the scores returned from each feature by the detector, where each score is the sum of the block scores for the corresponding feature. (Needs to be viewed in color.) ©2015 IEEE . . . 40
3.8 Three usual types of false positives, involving highly textured images, buildings and poles. 41
3.9 HOG, Hessian and LBP encode different properties of a 3×3 image patch. ©2015 IEEE 42
3.10 Example of a building image for which the HOG feature within our fusion detector responds false positively while the Hessian feature responds negatively. ©2015 IEEE 42
3.11 Example of a non-person image for which the LBP feature responds false positively while the Hessian feature responds true negatively. For reference, the two 3×3 arrays shown are the most common LBP patterns found in pedestrian images. ©2015 IEEE 43
3.12 (a):SVM weights when the Hessian first eigenvalue is negative. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect. 43

3.13 (b) to (e): Locations and magnitudes of the 1st eigenvalue which is negative for different orientation of the 1st eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 1st eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed [156] to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.) 44

3.14 (b) to (e): Locations and magnitudes of the 1st eigenvalue which is negative for different orientation of the 1st eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 1st eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.) 45

3.15 (a):SVM weights when the Hessian first eigenvalue is positive. The darker edge implies larger weight. The red circles highlight the regions where the Hessian and HOG weights are the most different (see figure 3.16). (b): Shape of the image surface that these weights try to detect. 46

3.16 Negative and positive SVM weights for HOG. The darker edge implies larger weight. Note that the edges have nine orientations. The red circles highlight the regions where the Hessian and HOG weights are the most different (see figure 3.15). 46

3.17 The corresponding regions (in red circles) of the HOG and Hessian weights are also highlighted in this example image. 47

3.18 (a):SVM weights when the Hessian second eigenvalue is negative. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect. 48

3.19 (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is negative for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.) 49
3.20 (a):SVM weights when the Hessian second eigenvalue is positive. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect. 50
3.21 (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is positive for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue positive in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.) 51
3.22 (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is positive for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue positive in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.) 52
3.23 Example images from the TUD-Brussel dataset. 55
3.24 Detection error tradeoff curve on the TUD-Brussel dataset. ©2015 IEEE . . . 55
3.25 Example image from the ETH sequence BAHNHOF dataset. 56
3.26 Detection error tradeoff curve on the ETH sequence BAHNHOF dataset. ©2015 IEEE 56

3.27 Example image from the Penn-Fudan dataset. 57
3.28 Detection error tradeoff curve on the Penn-Fudan dataset. ©2015 IEEE 58
3.29 Example images from the DaimlerChrysler dataset. 59
3.30 Detection rate – false positive rate curve on the DaimlerChrysler dataset. ©2015 IEEE 59
3.31 Example images from the INRIA dataset. 60
3.32 Detection error tradeoff curve on the INRIA dataset. ©2015 IEEE 61
3.33 An example where our detector is able to detect a pedestrian missed by the existing HOG+LBP detector. 62

4.1 Inferring pedestrian density (number of people) distribution in a building layout. Red spots are the possible regions that crowds gather. Floor layout taken from ION Orchard app. 64
4.2 Framework for developing and using our pedestrian density distribution model. 71
4.3 Diagram of our model showing the Markov relationships between the variables. 74
4.4 Model of a shopper moving from portals to portals. Portals are the shops, escalators and mall entrances. Numbers are the probabilities of finding a shopper in a node given a path ($P(\zeta|\eta)$). For simplicity of illustration, we assume that each node has the same span. 76
4.5 Example of three paths to choose from the origin portal to the destination portal. Each path has different probability. 77
4.6 Description of how each path descriptor is computed. *Viewdist* function returns the distance to the wall for an input direction. $Walkingdirection_i$ is the direction of walking from current node i to the next node. 78
4.7 Comparison of a conventional logistic function and our tanh-based likelihood function using an example in which there are two path descriptors: one that is linear (x-axis), and another that is cubic (y-axis). 80
4.8 The tanh function allows modeling of different rates of change of path preference with respect to path descriptor values. 81
4.9 Example of movements between vessels. $P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out})$ represents probability of finding a pedestrian moving between vessels from vessel categories $(\gamma_{in}, \gamma_{out})$. 84

4.10 Example of probability of finding a person moving from one vessel category to another vessel category. 86
4.11 Visual comparison of ground truth and predicted density distributions. Left of each pair: ground truth densities. Right of each pair: predicted densities. Further annotations: red circles are low volume escalators, yellow circles are high volume escalators, dark green circles are low volume mall entrances, light green circles are high volume mall entrances, and black circles are common corridors or intersections. Note that the numbers in the color legends do not directly map to density values, as they have been monotonically transformed to improve perceptual quality of visualization. (Has to be viewed in color.) . . . 98
4.12 Example where pedestrian density prediction can be improved by ramping up the popularities of escalators in circled regions. 101
4.13 Visualization of $P(\zeta)$ as the popularities of non-area categories are varied for Ion floor B4. Top: varying the popularity of low volume escalators. Bottom: varying the popularity of high volume escalators. The numbers below the layouts are the Pearson correlation coefficients of the generated crowd density counts and ground truth crowd density counts (shown on the left). 102
4.14 Log likelihood ratios when interpolating path preference parameters with varying amounts of priors. 106
4.15 Prediction on the Novena Square mall using parameters learned from the Ion mall. 107
4.16 Sigmoids of $P(\eta|\vartheta_{in}, \vartheta_{out})$ in (4.2) based on estimated path preference parameters. Red lines are sigmoids based on prior path preference values, while blue lines are sigmoids with the optimal interpolation of learned and prior path preference values. The interpolated sigmoids for (a) and (c) are degenerate horizontal lines because their interpolated thresholds, c'_j in (4.7), remain far to the left or right of the graphs. 109
4.17 Comparing density prediction of agent-based pedestrian simulation and our model prediction. Pedestrian simulation agents are much more unlikely to travel into the more remote regions (see (b)), where the pedestrian densities in remote regions are substantially under predicted, while more central regions are over predicted. Our model prediction (a) is closer to the ground truth in (c). 112

4.18 Comparing path probabilities of agent-based pedestrian simulation and our model between a fixed source sink. 113

5.1 Informal comparison of four shops in different categories. A food court (b) is often more crowded and has smaller person-to-person distances than a jewelry shop (a) which is often quite empty; the food court therefore has higher customer densities d_p. The stay time t_s in a hair salon (d), typically 30 to 60 minutes, is usually much longer than in a convenience store (c), where customers typically go to buy one or two items quickly. 117

5.2 An example comparison: Burger King and Tcc have different flow densities, measured at 0.35 and $0.05 pax/min/25m^2$ respectively. This is due to different stay times (customers tend to linger longer in cafes) and customer densities (fast food restaurants are typically more crowded). 118

5.3 Graphical comparison of four category popularity models with ground truth for Ion floor B2 and Novena Square floor 1. 127

A.1 Visual results (Left:Ground truths $P(\zeta)$ Right:Predicted pedestrian density distributions $P(\zeta)$ display on shopping mall layouts from learned **f** (see chapter 4.8.1) and path descriptor weights **a**, **b** and **c** (Need to be viewed in color, each $P(\zeta)$ is monotonically transformed to make comparison easier). 145

A.2 Graphical interpretation of p-value of Neyman Pearson Test. n_{pass} is the shaded area, and $n_{pass} + n_{fail}$ is the total area of null distribution 147

List of Tables

3.1 Description of each dataset, highlighting their differences and challenges. . . . 54

4.1 Shop categories used in this thesis, and corresponding archetypal stores. 72

4.2 The log likelihood ratios and p-values obtained in our experiments. Column (a) shows the test results obtained using the f_γ learned from the training sets, in an LOOCV procedure. Column (b) shows the test results with in-test optimization in which the best f_γ's are found for each individual test floor. 104

4.3 The log likelihood ratios and p-values obtained in our experiments in which we jointly learn from all floors in the Ion mall and test on each floor in the Novena Square mall. Column (a) shows the test results obtained using the f_γ learned from the training sets, in an LOOCV procedure. Column (b) shows the test results with in-test optimization in which the best f_γ's are found for each individual test floor. 108

5.1 Prediction thresholds (maximum likelihood ratios) using learned q_γ (learned flow densities) and their p-values. 121

5.2 Collected customer flow data of every shop in Ion floor B2. Columns 1 (Shop Name) shows the name of each shop vessel. Column 2 (Vessel Category) shows the vessel category of each vessel. Column 3 (Area) shows the area of each shop vessel. Column 4 (In) shows the number of people moving into each shop vessel over the period of time specified in column 7 (Duration) and likewise column 5 (Out) shows the number of people moving out of each shop vessel over the same period of time. Column 6 (Mean in and out) is the mean of column 4 (In) and column 5 (Out). Column 8 (Measured Flow/min) is the mean number of people moving in and out of each shop per minute. 123

5.3 Collected customer flow data of every shop in Novena Square floor 1. Columns 1 (Shop Name) shows the name of each shop vessel. Column 2 (Vessel Category) shows the vessel category of each vessel. Column 3 (Area) shows the area of each shop vessel. Column 4 (In) shows the number of people moving into each shop vessel over the period of time specified in column 7 (Duration) and likewise column 5 (Out) shows the number of people moving out of each shop vessel over the same period of time. Column 6 (Mean in and out) is the mean of column 4 (In) and column 5 (Out). Column 8 (Measured Flow/min) is the mean number of people moving in and out of each shop per minute. 124
5.4 Total variation distances between four category popularity models and ground truth for Ion floor B2 and Novena Square floor 1. 128

List of Abbreviations

LIDAR	Light Detection and Ranging
GPS	Global Positioning System
GIS	Geographic Information System
CENTRIST	CENsus TRansform hISTogram
LSA	Latent Semantic Analysis
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
SVM	Support Vector Machine
PCA	Principal Component Analysis
EM	Expectation-Maximization
GT	Ground Truth
LOOCV	Leave-One-Out Cross Validation
DIY	Do It Yourself

List of Notation

$\text{mag}_{j,i}, \text{ang}_{j,i}$	Magnitude and angle of gradient at pixel location (j,i)
$dx_{j,i}, dy_{j,i}$	Gradient along x and y axis respectively at pixel location (j,i)
$I_{j,i}$	Image pixel value at location (j,i)
$dxx_{j,i}, dyy_{j,i}, dxy_{j,i}$	Second order gradient (curvature) in x, y axes respectively at pixel location (j,i)
$\mathbf{H}$	Hessian matrix
$\mathbf{u_1}, \mathbf{u_2}$	Eigenvectors of Hessian matrix
$s1, s2$	Eigenvalues of Hessian matrix
$P(\zeta)$	The probability of finding a person in a node ζ on a floor layout
$P(\zeta\|\eta)$	The probability of finding a person in a node ζ if it is known that the person is taking the path η
$P(\eta\|\theta_{in}, \theta_{out})$	The probability of a person taking a particular path η given all possible paths between origin portal θ_{out} and destination portal θ_{in}
$P(\theta_{out}\|\vartheta_{out})$	The probability of a person leaving a vessel ϑ_{out} through portal θ_{out} and likewise $P(\theta_{in}\|\vartheta_{in})$ is the probability of a person entering a vessel ϑ_{in} through portal θ_{in}
$P(\vartheta_{in}, \vartheta_{out}\|\gamma_{in}, \gamma_{out})$	The probability that the source vessel is ϑ_{out} and the sink vessel is ϑ_{in} if we only knew that these vessels belong to the categories of γ_{out} and γ_{in} respectively
$P(\gamma_{in}, \gamma_{out})$	The probability of finding a person moving from a vessel of category γ_{out} to another vessel of category γ_{in}
$\text{span}(\zeta)$	A function that returns the longest principal length of the node (based on Principal Component Analysis (PCA))
$\mathscr{Z}_\eta$	Set of all nodes in path η

$\mathscr{H}$	The set of paths start and end with portals θ_{out} and θ_{in}
$\mathscr{V}_\gamma$	The set of vessels belonging to vessel category γ
Θ_ϑ	The set of portals belonging to vessel ϑ
β	Interpolation parameter for interpolating path descriptor parameters with prior
$\text{desc}_j(\eta)$	Descriptor j of path η according to the descriptor sequence (Expanse, line of sight, distance and turn distance) and $j = 1,\ldots,4$ respectively
a_j, b_j, c_j	Parameter of path descriptor j
$\text{area}(\theta), \text{area}(\gamma)$	Area of vessel ϑ or vessel category γ
$\text{cat}(\vartheta)$	Category which the vessel ϑ belongs to
$\mathscr{S}$	Set of all vessel categories that each contains only one vessel
$K_1(\gamma_{in}, \gamma_{out}), K_2(\gamma_{in}, \gamma_{out})$	Functions to redistribute popularities in single-vessel categories to other vessel categories
$f_\gamma, \mathbf{f}$	Popularity of the non-area vessel category γ, vector of f_γ
m_i	Number of pedestrians in node ζ_i
m_i'	Ground truth of pedestrian count in node ζ_i
p_i	$P(\zeta)$ of node i
T	Number of people in a floor
N	Number of nodes in a floor
$L(\mathbf{m}'; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{f}_\gamma)$	Log likelihood function given ground truth pedestrian counts
λ	(Negated) joint log likelihood function using normalized ground truth counts
R	Log likelihood ratio to compare our model prediction with baseline prediction
$\mathbf{s}$	A sample from baseline uniform multinomial distribution
t	Log likelihood ratio of generating ground truth pedestrian count $\mathbf{m}'$ from baseline and model distributions
n_{pass}, n_{fail}	Number of samples drawn from baseline uniform multinomial distribution that pass or fail the Neyman Pearson test
ρ_k	p-value of test k
ρ_{meta}	Combined p-values using Fisher's method
$\text{flow}(\vartheta), \text{flow}(\gamma)$	In out rate of vessel ϑ or all vessels in category γ
$t_s(\gamma)$	Stay time in shop vessel
κ	Number of parallel queues in a shop

$d_p(\gamma)$	Customer density in a shop
q_γ	Flow density, number of people moving in and out of vessel category γ per second per area

Chapter 1

Introduction

1.1 Background

Many factors and considerations are involved when it comes to planning new urban spaces or redesigning old ones. One key consideration and challenge is in predicting how people will move about in a new urban layout, as they go about their daily activities. If these movements can be modeled accurately, then urban designs iterated towards an optimally balanced combination of utility, efficiency, aesthetics and experiential harmony.

Predicting the movement of individuals over an extended period of time is generally impossible, due to the large number of unknown factors and great variety of potential actions, leading to a chaotic system. However, if the problem was approached from a statistical perspective, then it becomes amenable to modeling. We may not be able to predict with any certainty whether a particular person will walk through a particular point of space at a particular time, but it may be possible to state that that on average a certain number of people may be presented in some specific region of space at any given moment, or the average number of people that is expected to pass through a specific doorway over a fixed time interval.

It is not straightforward how such a model can be created. What has become more apparent in recent years is that hand crafted models, based on human intuitive reasoning, do not perform

as well as data-driven models that are learned from a large volume of observations.

Indirect data sources on pedestrian movement include geolocated social media posts (e.g. tweets) and travel surveys. These are however sparse and imprecise, with little control over the intended spatial regions of interest. Alternatively it may be possible for individuals to contribute extensive GPS tracks through specialized apps on mobile devices, but likewise the observations are often limited due to greater effort needed and privacy concerns.

The other mainstream avenue for gathering substantial data of pedestrian movement is via cameras, which is the approach taken in this thesis. Cameras have the advantage of a small footprint but capturing a large area, and from a practical perspective there are already many existing surveillance cameras installed.

In the first part of this thesis, we will look into the problem of sensing pedestrians, more specifically detecting people in general images. To get a sense of the output from the pedestrian detector that is developed in this thesis, see figure 1.1.

In the second part of the thesis, we address the problem of creating a model for predicting the steady-state pedestrian density distributions in a urban layout. The outcome of the model that is eventually developed in this thesis may be visualized as shown in figure 1.2.

Crowd density modeling and prediction within buildings can help maximize the utility of space for physical, social and economic purposes. For example, it may facilitate optimal zoning of shopping malls for different shop categories and other facilities. It may predict the congested and underutilized regions in a new mall design and allow for appropriate setting of rental prices in different locations of the mall. It can aid in planning for events and promotions when larger crowds are expected. It can also support future mall renovations by indicating better locations for additional shops and escalators.

Figure 1.1: Detection results from our pedestrian detector. Purple boxes are the false positives. In this figure, a correct detection is recorded if the detected bounding box intersects the ground truth bounding box by more than 50% of the total combined area of the two bounding boxes, following established norms for experimental evaluation (see chapter 3).

1.2 Motivation

The first part of the thesis is focused on detecting pedestrians in images, more specifically determining the positions and sizes of people in camera images.

This is challenging as there is great variability in the content of images, arising from large differences in lighting, background clutter, variations in clothing, pose, body size and hairstyle, and occlusion. Our detection algorithm needs to be robust against these problems.

Although there are commercial systems such as Mobileye [3] and BMW [1] that can detect pedestrians, it remains a difficult problem under more unconstrained environments. Vehicular systems are typically more concerned about the immediate surroundings and obstacles, rather than distinguishing between people and other objects. Conversely, the detection and tracking solutions needed for urban informatics require a different approach, with the focus on maximizing classification accuracy.

The second part of the thesis focuses on predicting the density distribution of pedestrians in

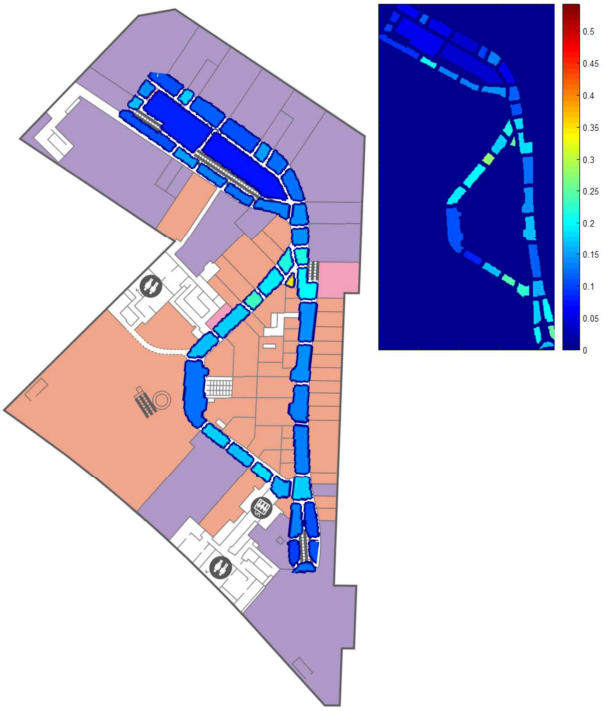


Figure 1.2: Inferring crowd density distribution given a building layout. Differences in color hues mean different density distribution values as shown in the color bar. Top right is ground truth. Floor layout taken from ION Orchard app.

a given floor layout design of a mall. Even if we avoid the impossible task of predicting the extended individual movement of all pedestrians, it remains challenging to accurately model the collective dynamics of crowd behavior. Unlike fluid dynamics, there are no established models that work over an extensive period and in general scenarios.

The work in this thesis is limited to modeling only the pedestrian density distribution at an appropriate timescale for which it may be reasonable to assume that the distribution is stationary, i.e. we assume that the distribution does not change over time. We expect this to correspond roughly to the distribution of people averaged over an hour, during a non-peak mall operating hour. In particular, we do not model at such a fine temporal granularity that the distribution changes as groups of people move about, nor do we model the transitions from peak to non-

peak hours.

Even with these limitations, the problem remains difficult. For example, shoppers obviously do not just spread out evenly across the floor to fit a naive uniform density model, as may be seen from the ground truth count visualization in figure 1.2. These variations in densities are due to various reasons. Some shops are more popular and will receive more traffic, and shoppers moving from one shop to another may have a choice of different routes. Two of the questions that will be explored in this thesis are: can we predict in advance the relative popularities of different shops (or types of shops), and can we figure out the factors influencing how much shoppers prefer one route to another if both lead to the same destination?

While people are diverse in nature, we nonetheless expect that there are some commonalities in their behaviors that are predictable, as least in a probabilistic sense. These commonalities may for example lead to one route being more popular than another. However, predictability requires that the underlying factors be discovered. Do people on average prefer a shorter path or a longer one with fewer turns, or yet another that passes through open spaces and is thus less claustrophobic? How do these factors interplay relative to each other? Instead of trying to intuit these relationships, we will try and arrive at these answers from a data-driven approach.

We also want an approach that can ideally have a fast enough implementation to allow for an urban planner to interactively change designs and see the resultant density distributions rapidly. We expect that an agent-based approach [48] to pedestrian simulation is too computationally costly to achieve this goal, as it involves simulating the movement of every single person. These simulations lead to nice animations that are useful for entertainment as well as for intuitively understanding the critical problems in scenarios such as emergency evacuation. Our intention is instead to come up with a model that is not too computationally complex, able to learn from data and at the same time accurate enough for modeling of steady state crowd densities.

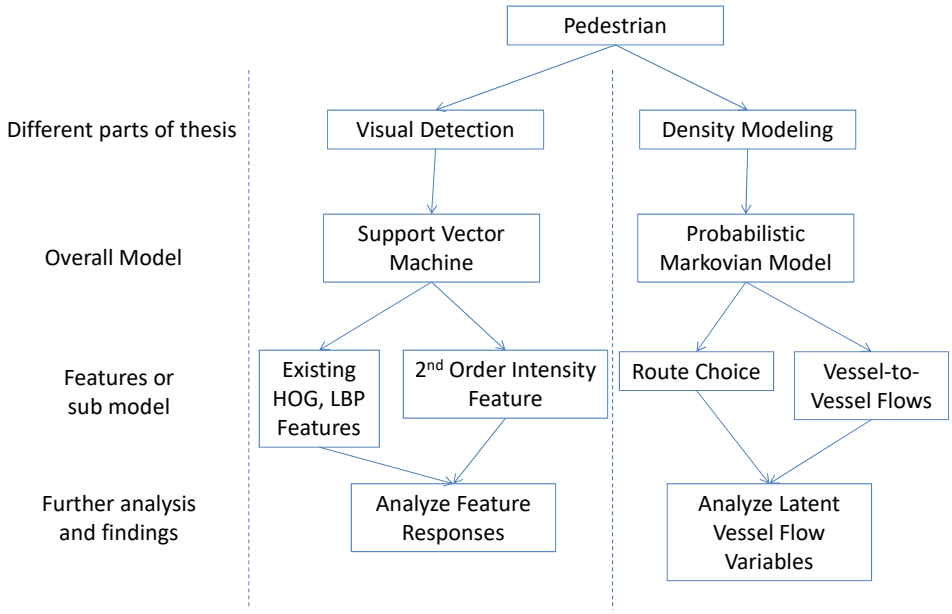


Figure 1.3: Thesis framework

1.3 Research Objectives and Contributions

The thesis framework is shown in figure 1.3. The two major directions in the thesis are detection of pedestrians in camera images, and pedestrian density distribution modeling for floor layouts of malls.

The first objective is pedestrian detection, for which the following are contributions in this thesis:

- We proposed a new feature which captures second order intensity variation. We have shown that when this feature is combined with HOG (Histogram of Oriented Gradients) [33] and LBP (Local Binary Patterns) [156], it leads to more than 10% improvement in pedestrian detection accuracy in most datasets. Use of these features is also more computationally efficient in inference and learning than deep learning approaches.

This enables implementation in low power embedded systems and operates in interactive time.

- We presented various novel insights from carrying out in-depth analysis on failure modes of previous features. We highlighted new patterns that our new feature is capable of recognizing by detailed visualization of the responses from our Support Vector Machine (SVM) detector. We also analyzed the weights of corresponding features in the SVM to see how they adapt when our feature is combined with other features.

A second objective is to develop a model for predicting crowd density distributions in a shopping mall. The contributions are as follows:

- We proposed a highly novel probabilistic model for predicting the approximately steady-state pedestrian density distribution in a shopping mall. The model establishes the Markovian relationship between different latent variables and parameters.

- To our best of knowledge, this is the first time route choice modeling is attempted based on pedestrian count data instead of trajectory histories from tracking instruments such as GPS. The preferences of different routes are hypothesized to be dependent on a number of path descriptors inspired by space syntax theory [67] in architecture, and the route choice model is designed to learn trade offs between these path descriptors. We showed that our model was able to achieve practical accuracy on a shopping mall dataset, fitting the ground truth pedestrian counts better than the baseline objective prior of a uniform multinomial distributions and was also significantly more accurate and faster than a baseline system using agent-based simulation [48].

- We further proposed modeling the shop popularities as latent variables that affect the pedestrian density distributions, but which are themselves influenced by category-specific popularity parameters. We showed that despite learning only from ground truth pedes-

trian counts, the inferred latent shop popularities correlated well with the measured flow rates into and out of shops.

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 contains a literature review of existing pedestrian detection methods and pedestrian density distribution models. Some validation techniques for pedestrian density distribution models are also covered. Chapter 3 describes our pedestrian detection algorithm, in-depth analysis and some experiment results on existing datasets. Chapter 4 introduces our probabilistic Markovian pedestrian density distribution model, and in addition covers aspects related to data collection, statistical validation and experimented results. Chapter 5 describes modeling shop popularities and provides further experimental results. Chapter 6 concludes the thesis, summarizing key contributions and describing our future work.

Chapter 2

Literature Review

2.1 Crowd Sensing

Various sensing techniques can be used to determine the presence and distribution of pedestrians. These include manual, electronic, and computer vision techniques, which are reviewed in subsequent sections.

2.1.1 Manual Counting

Researchers from Space Syntax Limited have developed techniques [148] for counting pedestrians in streets and buildings. One way is to draw a line physically on the ground and have a field statistician count the number of people crossing the line in both directions. A second method is to observe people at walkway intersections and count the number of people moving from one walkway to another. A third method involves traversing a building and recording the number and the locations of people spotted. A fourth method is to trail individual pedestrians to determine the route each pedestrian takes, but this is very labor-intensive and may influence the natural behavior of the pedestrians.

2.1.2 Global Positioning System (GPS) and Travel Survey

Pedestrian travel patterns may be obtained using GPS trackers or mobile phones [80]. Alternatively, travel surveys may be conducted to sample travel patterns including journey start and end times, origins and destinations, as well as the routes taken.

Methods that automatically count people in video can provide a denser sampling than GPS tracking. While GPS tracking requires active participation by pedestrians who need to keep their GPS trackers turned on, automated video analysis can sample pedestrians passively. Mobile phone tracking may be noisy, requiring Wi-Fi and accelerometers to further improve accuracy. In addition, there may be privacy concerns when collecting extensive unfiltered travel patterns.

2.1.3 3D Sensing

Range sensors such as LIDAR (Light Detection and Ranging) and Kinect can be used to not only detect the positions but also the heights of pedestrians. The heights may be used to differentiate children from adults. Brscic et al. [27] use range sensors mounted on the ceiling to detect humans in a shopping mall.

2.1.4 Image-Based Pedestrian Sensing

In this section, we review existing computer vision approaches for pedestrian detection, tracking in crowds, activity analysis and crowd modeling.

2.1.4.1 Pedestrian Detection

There are three main approaches to pedestrian detection. They are background subtraction, motion across frames and sliding windows with appearance information.

Background Subtraction One category of techniques based on background subtraction involves modeling humans as foreground blobs [178, 126, 54] (obtained after background subtraction). Pedestrians are then extracted through segmentation. The advantage of this approach is that background subtraction is simple to implement. However both the camera and background have to be static and it is difficult to segment large foreground blobs consisting of groups of pedestrians into individuals.

Another category involves detecting humans using multiple cameras and ground plane constraints [74, 43, 55, 167]. In [74], Khan and Shah first obtained a foreground likelihood map in the viewpoint of each camera and used homography constraints of the ground plane to locate the feet of each person. Images of the feet are stacked together to form a space-time volume. The graph cut method is then applied to segment the volume into feet trajectories. Alternatively, heads may be detected instead of feet [43]. The advantage of this category is that multiple cameras provide robustness to occlusion when pedestrians are close together, with the drawback being that it needs multiple distributed cameras covering the same area.

Motion Across Frames This approach uses motion across image frames. Pedestrians are detected by clustering a set of features tracked across frames [144, 26]. In [144], Sugimura et al. cluster features based on similarity in terms of gait features, local patch appearance, coherency and spatial proximity of features. Although easy to implement since detection is based only on simple motion tracking, a drawback is that a group of people moving in the same direction is difficult to segment based purely on motion. In some instances, motion features may not be extracted on pedestrians who are wearing uniform textureless clothing. The method in [170] is similar to [144], except that Yu et al. not only cluster the features based on motion trajectories, but also using local appearance information (using cookbook of intensity shapes) around the feature points, and co-occurrence relationships of these features by local appearances.

Sliding Windows with Appearance Information The main pedestrian detectors that we are concerned with in this thesis are those based on sliding windows. These detectors have high accuracy which are suitable for many applications.

Rodriguez et al. [127] detect heads of pedestrians using image gradients and adjust region-specific thresholds for head detections using a crowd density map estimated from the image.

Some detectors use more advanced appearance features. In [33], histograms of oriented gradients (HOG) are used. The idea is that shapes can be represented quite well by the distribution of local edge directions, without the need to know exact edge positions. In [46, 47], the framework consists of a root filter and a number of part models. In [85], local features of humans are extracted around interest points and clustered to form a codebook. For a codebook entry, the spatial occurrence distribution is stored as a list of occurrences. In [147], covariance matrices are used as descriptors. Haar wavelets are used as features in [116], while [161] used the CENTRIST descriptor. In [131], shapelet features are used. In [41], feature mining is conducted to look for good features in a set of generalized Haar features for pedestrian detection. In [92], Maji et al. use an intersection kernel support vector machine to do pedestrian detection on multi-level histograms of oriented edge features. In [158], Wojek and Schiele combined a few features to produce a better detector. In [153], Walk et al. created a new feature based on self-similarity of color channels to do pedestrian detection. In [18], Bar-Hillel et al. developed a new feature selection algorithm called predictive feature selection for pedestrian detection. In [40], Dollar et al. developed integral channel features which were fast to compute for pedestrian detection. In [38], Dollar et al. noticed that most of the computation cost of their integral channel features detector [40] laid in image resizing for constructing an image pyramid and feature computations at each layer of the pyramid. They sped up detection by using an image pyramid with fewer layers and used interpolation to obtain feature values for other layers. In [20], Benenson et al. avoid rescaling the image to multiple sizes and did detection at one single

scale by using multiple detection models, each model for each image size. Instead of using N models where each model is used for detection for each scale, they reduced the number of models from N to N/k by interpolating the weights of the models between different scales to get additional models in between the scales. Besides appearance features, motion cues are also used in [34, 150].

Some of the detectors use shapes for detecting humans. In [53], human shapes are detected through chamfer matching. In [89], Lin and Davis detect human shapes by parts and then combine the result using a probabilistic model.

Some detectors use segmentation and appearance features. In [112], Ott and Everingham use segmentation of local patches to extract gradient features. In [87], Leibe et al. use segmentation and chamfer matching to verify detection.

One class of pedestrian detectors incorporates depth cues to detect humans. In [44], appearance and depth cues were used, while [52] exploited depth, shape and texture. In [19], Benenson et al. developed a fast way to compute a low detail depth map and used that to speed up pedestrian detection.

Based on the analysis in [42], the HOG-LBP detector [156] which uses local binary patterns (LBP) in addition to HOG, achieves the best performance when comparing per-window results.

The advantage of the sliding windows approach is that it does not require multiple cameras and can do detection on a single frame of video. The accuracy is quite high as long as people are not standing too close to each other and their legs are not occluded. The drawbacks are that it needs the torsos and legs to be visible for detection and it is computationally expensive as it requires scanning all possible shifts and scales of the detection window.

Newer techniques based on sliding windows include variations of integral channel features [40] and deep learning [90]. More recently, a deep learning approach to feature extraction for

pedestrian detection has shown good promise [90]. This work learns a complex hierarchical feature, saliency detection and body parts detection in one deep network. However, the use of such networks is costly — in order to reduce computational time, Luo et al. [90] use an SVM in a first pruning pass for both training and testing. Deep learning approaches have a risk of overfitting [105] to a particular dataset due to unintended but systematic differences in lighting and backgrounds across different datasets. Haar-like features [177] improve on integral channel features. Instead of using a single rectangle integral to make decisions in a boosting tree, it uses a combination of multiple rectangles of different sizes and positions.

Costea and Nedevschi [32] used patch dictionaries, while Watanabe and Ito [157] utilized co-occurrence histogram features.

2.1.4.2 Tracking Pedestrians in Crowds

There are five main approaches for tracking pedestrians in crowds, namely tracking by blobs, tracking by motion of features, tracking using multiple cameras and tracking by detection.

Tracking by Blobs In [128], Ryan et al. extract blobs in an image using foreground segmentation. The blobs are tracked through split and merge. The limitation is that it does not split the group blobs into individuals.

Tracking by Motion of Features In [144], Sugimura et al. used a KLT tracker [138] to generate a set of trajectories. Stationary trajectories (i.e. corresponding to background features) and spurious trajectories are eliminated. A graph is generated using Delaunay triangulation of points from the remaining trajectories at every time step. The edges of the graph are assigned weights to represent dissimilarity. The dissimilarity is measured in terms of gait features, local patch appearance, coherency and spatial proximity of motion. To obtain the gait features, they

first use linear regression to convert the trajectory from two dimensions to one dimension and extract the periodic component. Then they apply Fourier transform to find the amplitude and phase of the trajectory. Local appearance is determined by the change in visual appearance over time of a local triangular patch formed by points in each time frame on three adjacent trajectories. The graph is segmented into connected sub-graphs and each sub-graph represents a person's trajectory. The limitation is that tracking can fail due to lack of reliable trajectories and multiple clusters can occur in one person due to corresponding trajectories having different frequencies.

For tracking people in high density crowds, Ali and Shah [9] use optical flow to extract the directions of crowd movement. They then detect the boundaries of the scene by segmenting based on the directions of optical flow in the video. The detected boundaries are used to improve the prediction of the direction of the movement of each person in the video. The drawback of this technique is the difficulty of track initialization as the people in the video are typically small.

Tracking using Multiple Cameras In [74], Khan and Shah first obtain a foreground likelihood map from the view of each camera. This is done by modeling the background with a mixture of Gaussians. They use the homography constraint of the ground plane to locate the feet of people visible in each camera. The feet images are stacked together to form a space-time volume, which is then segmented by graph cut into feet trajectories. One limitation of this method is susceptibility to shadows.

Instead of detecting feet, Eshel and Moses [43] detect each person's head. They use simple background subtraction, after which the tops of heads are detected by analyzing planes at different heights and finding the highest plane for each person, constrained by planar homography.

The limitation here is that it requires multiple cameras aimed at the same area but from different directions, thus needing a more elaborate set up. Person re-identification [6] may be applied across cameras through consistency of appearances of clothing and hair style.

Tracking by Detection The system proposed by Ess et al. [45] is based on a pair of cameras. For each frame, a depth map, ground plane and camera pose are predicted, after which human detection is conducted. Tracking is carried out using a graphical model, and the tracked locations in world coordinates are used for visual odometry calculation in the next frame.

In [33], Dalal and Triggs use histograms of oriented gradients for detecting pedestrians. It has a high detection rate but it requires the full body to be seen and it is computationally intensive.

For data association between different frames, some methods incorporate predictive motion models. In [122], Pellegrini et al. assume that people will try to avoid colliding with each other when walking and use this constraint for prediction. In [165], besides modeling collision avoidance behavior, Yamaguchi et al. also model group behavior in which people in a group will usually maintain the same walking speed relative to each other. In [83], Leal-Taixe et al. try to predict the movements of people using group and social behavior models. In [109], Okuma et al. use boosted particle filter. They detect object hypotheses using Adaboost and use filtering process to keep track of individual objects. In [86], Leibe et al. combine the estimation of trajectories and detection as a single optimization problem. In [176], Zhang et al. formulated the data association problem of classifying the positions, scales and appearance of each object into non-overlapping trajectories as a cost-flow network problem. In [139], Shitrit et al. conduct multiple objects tracking using a data association method, based on solving a convex optimization problem, that is effective even when the appearance cues are at distant time intervals. In [172], Zamir et al. try to link all detection instances belonging to the same person through generalized minimum clique graphs within which the costs of the links are based on motion and appearance features.

Methods that include all the frames for data association tend to do better than methods that try to associate detections only across pairs of consecutive frames. However such methods are more computationally expensive.

2.1.4.3 Activity Analysis

The three classes considered here are the space-time, sequential and hierarchy approaches. A review of activity analysis can be found in [5].

Space-Time Approach A space-time volume may be created by stacking time sequential images. Bobick and Davis [23] use a 3D binary motion volume, created by stacking thresholded foreground images over time created from background subtraction. A motion image is created from the 3D space-time volume using weighted 2d projection of the volume across the time axis. Space-time volume correlation [137] may be used, in which each local volume captures a particular local motion and can be correlated with a template to give a local match score. By aggregating these scores, the overall correlation is computed. In [73], Ke et al. segmented the spatial-temporal volume. Recognition is done by matching the spatial-temporal volume with a shape of action model. The limitation of using 3D binary motion images is that it requires a background image for background subtraction. The sequence of foreground images can capture the action of a single person quite well, but if there are more than one person located in proximity, the foreground obtained is made up of one joint region instead of separate regions for each person, and therefore individual actions cannot be recognized. Space-time volume correlation is a very expensive method and is prone to error in changing lighting conditions. When segmentation is attempted on the spatial-temporal volume, the result is often poor as it is a difficult problem.

Other approaches [81, 39, 107, 169, 130] extract interest points from the space-time volume. These points are robust to small movements of the camera, background movements, changes in

illumination and noise. For each point in a space-time volume, Zelnik-Manor and Irani [173] calculate a normalized intensity gradient. They use these space-time features to create a histogram, then matching is done by comparing the distances between histograms. In [58], Gorelick et al. use the Poisson equation to extract local shape properties of a space-time volume. Niebles et al. [107] use Latent Semantic Analysis (LSA) to recognize actions. The limitation is that the interest points are sparse in the space-time volume and therefore the detection rate is often low.

To get spatiotemporal proximity information from features, Savarese et al. [133] use feature co-occurrence patterns in the local space-time region. Laptev et al. [82] divided the whole space-time volume into a grid of cubes and computed a spatiotemporal histogram for each cube. The drawback is that the histogram is expensive to compute.

Sequential Approach This approach treats the input as a sequence of feature vectors. A dynamic time warping algorithm can be used for matching two sequences [35, 51, 149]. In [164], Yacoob and Black apply principal component analysis to create a set of activity bases. Then they use a linear combination of activity bases to represent an activity. Hidden Markov Models (HMM) can be used to recognize activities [166, 142, 22] as they are able to represent feature changes during the activities reliably. Oliver et al. [110] use a Coupled Hidden Markov Model to model interaction between two people, while Park [117] uses a dynamic Bayesian network instead. Dynamic time warping is an expensive operation as it has to match the current sequence to all the sequences in the database to find the best match. HMMs are good at discovering very likely sequences of feature vectors. Both dynamic time warping and HMMs are not able to detect recursive patterns, unlike methods based on context free grammar [70, 98, 96].

Hierarchical Approach A hierarchical approach recognizes high-level activity by first recognizing low-level activities (sub-events). The advantage is that it is able to recognize struc-

turally more complex high-level activities. Hierarchical models require less training data than single layer models. Hierarchical models also make it easier to incorporate prior knowledge. Some researchers use multi-layer HMMs [110, 106, 175] while others use statistical context free grammar [70, 98, 96]. The temporal relationship among sub-events can also be modeled [11, 12, 123, 140, 152, 129, 59]. Some researchers use Petri nets [171, 102, 56] to represent and recognize human activities.

2.1.5 Image-based Crowd Analysis

Crowd analysis, in the context of this section, covers estimation of the densities and sizes of crowds, as well as crowd behavior understanding. For density estimation, we will only review methods that infer crowd densities directly from images, as opposed to simple counting after individual pedestrian detection (which has been discussed previously). A review of crowd analysis can be found in [71].

2.1.5.1 Density Estimation

Davies et al. [36] use background subtraction and edges to estimate the sizes of crowds, while Regazzoni et al. [125] use the total contrast strength of the vertical edges. Their results were better than another method using belief networks [113]. Ma et al. [91] estimate crowd size by counting the number of foreground pixels labeled as a crowd. However the linear model used is affected by occlusion which is also the case in [36]. Kong et al. [76] use histogram of object areas and edge orientations as features, normalized to the camera's perspective.

Marana et al. [93] use gray-level dependence matrices as texture features which input to a Bayesian classifier that outputs five different levels of densities. Wu et al. [162] use multi-resolution gray-level dependence matrices as features and an SVM as classifier. They achieved a maximum error below 5%. Chan et al. [28] apply texture-based motion segmentation to a

crowd. For every motion segmented cluster, foreground segment features, edge features and texture features are extracted. These features are fed into a Gaussian process to determine the number of people. Chan and Vasconcelos [29] used a similar idea except that Bayesian Poisson regression was employed to find the size of a crowd. Both methods [28, 29] produce good results although it very much depended on the accuracy of the segmentation.

2.1.5.2 Crowd Behavior Understanding

One approach to crowd behavior understanding is to consider the interactions of different person-level tracks. Jacques et al. [72] track people in a top-down camera and use a Voronoi diagram to cluster the people into groups. Cheriyadat and Radke [31] segment low-level motion features into individual trajectories, and the longest common subsequence is used to identify dominant movement. Unusual individual motion is detected if it is poorly aligned to the dominant motion. Wang et al. [155] use a hierarchical Bayesian model to connect visual features, activities and interactions.

At a coarser granularity, Boghossian and Velastin [24] do motion estimation using a block matching algorithm. By using flow paths and directions, they can detect congestion near exits (circular flow near the exit due to many inter-person collisions between opposing flows), fire outbreak (outward radial flow), and injured persons (obstacles in flow paths). Ali and Shah [8] detect crowd instabilities using Lagrangian particle dynamics. Mehran et al. [94] detect abnormal behavior of crowds by using a social force model. Kratz and Nishino [77] divided a video into spatial-temporal cubes and the spatial-temporal gradient was calculated for each cube. A HMM is used to model spatial and temporal changes between cubes. They reported good results for anomaly detection, but one drawback is the difficulty in determining the size of the cube to use in the presence of perspective distortion.

The collective properties of crowds include density, speed and direction. Crowd density and velocity can also be used to detect an abnormal event when there is a sudden change in those

measurements. There are also more complex behaviors, such as the formation of groups, lanes, and counter flows. Shao et al. [136] try to determine various attributes in videos of crowds, by modeling the collectiveness, uniformity, stability and conflict of crowds. Shao et al. [135] learn type and location attributes of crowd videos such as indoor, performance and conference. Yi et al. [168] model the stationary behavior of crowds typically found in the waiting areas of train stations.

2.2 Crowd Modeling

Apart from sensing crowds, we are also interested in crowd modeling. Crowd models provide a means of predicting various crowd properties in different environments and scenarios. Crowd modeling spans multiple fields, such as computer vision, transportation, pedestrian simulation, sociology, psychology and urban design.

Crowd models can be classified as macroscopic, mesoscopic and microscopic. Macroscopic models treat a crowd of people as a whole, while microscopic models work at a level of individual pedestrians in a crowd. Mesoscopic models lie in between. A review of crowd models can be found in [174].

Some researchers use physics inspired models. Helbing [62] uses fluid dynamics to model crowds at a macroscopic level. Helbing and Molnar [66] use a social force model to model crowds at a microscopic level. Some researchers use agents [15, 101, 115] to model individuals in a crowd. Cellular automation [134] is another approach. In [79], the floor area is divided into cells. Each cell can represent a person, a wall or a free floor area. Pedestrians move between cells following predefined rules. Nature inspired models are also a possibility. Banarjee et al. [17] use an emotional ant model, with an ant colony optimization algorithm. Kirchner and Schadschneider [75] model interactions in a crowd using ideas from chemotaxis, which relates to the movement of an organism in response to chemical stimuli.

2.2.1 Pedestrian Simulation

Pedestrian movement may be simulated via agents [48], fluid dynamics [69], cellular automata [134], and social force modeling [66]. Helbing and Molnar's social force model uses the equation:

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0 - v_i}{\tau_i} + \sum_{j \neq i} f_{ij} + \sum_W f_{iW} \tag{2.1}$$

which broadly states that the difference in desired velocity v_i^0 and current velocity v_i must tally with sum of external forces f exerted by surrounding nearby agents and obstacles. These simulations are often used for modeling the evacuation of pedestrians [63] in buildings, train stations, and airplanes. Pedestrian simulation can also be used to evaluate the design layout of a city [14]. There are also long and short term models in pedestrian simulation [179].

Pedestrian simulation is used in areas such as military simulation, evaluation of architecture design, safety engineering (e.g. fire evacuation), computer graphics animation and social studies (investigating how opinions and extreme ideologies are formed) [179].

Pedestrian simulation is computationally expensive as all individuals have to be simulated.

2.2.1.1 Validation Techniques for Pedestrian Simulation

Validation of pedestrian simulation is a difficult problem because pedestrian behaviors are sensitive to many different factors, not all of which can be observed or modeled easily. Validation techniques are discussed in this work [103]. One method is facial validation [121], in which the simulation output is assessed subjectively by subject matter experts. Another would be empirical validation [25], in which assessment is done through statistical tests or quantitative measurements.

Behavior validation [60] can be used to assess whether a model is able to produce a similar result given similar input (either internal or external) to the real system. For example validation

can be done by comparing a group of pedestrians standing in different locations and walking in different directions will move in such a way that they have the same collision avoidance behavior as the model. Structural validity [97] mainly relies on the use of facial validation to compare a model to assumptions, theories or other research fields such as psychology or social science.

Sensitivity analysis [132] is also usually carried out on pedestrian simulation. Researchers are interested in knowing how much the output would change when there is a small perturbation in the parameter values.

Validating a pedestrian simulation model is a complex and time-consuming process. Often, a pedestrian simulator is only partially validated.

2.2.2 Trip Modeling

2.2.2.1 Route Choice Modeling

When there are multiple paths to the same destination, it is useful to figure out which paths pedestrians will select. This can be modeled by a route choice model [124] based on transport theory. Such a model may use information like lengths of paths to produce a path preference given by

$$P_k = \frac{\exp(V_k + B_{CF} \times CF_k)}{\sum_{i \in C} \exp(V_i + B_{CF} \times CF_i)} \tag{2.2}$$

where P_k is the probability of choosing route k within a set of paths C. V_k and V_i are the utilities of paths k and i. The utilities are the raw path preference values computed from path descriptors such as the path length. CF_k and CF_i are commonality factors that reflect how the utility of a path is affected by the presence of other paths. B_{CF} is the parameter to be estimated. Estimation of the origin-destination matrix [13] can also be used to model the frequency of movement between a set of source and destination pairs.

A route choice model differs from a fluid-based pedestrian model in that the choice of movement can be affected by non-local factors. This caters for the more valid assumption that humans can reason and plan ahead, whereas the movement of a fluid particle at each moment is based only on its immediate environment.

It is possible to use a gravity model [151] to represent the volume of movement between origins and destinations given the population sizes of source and destination regions:

$$T_{ij} = K_i K_j T_i T_j f(C_{i,j}) \tag{2.3}$$

where T_i and T_j are the number of trips between i and j locations, $C_{i,j}$ is the travel cost and K_i and K_j are the balancing factors. There is a non-linear function f applied to the travel cost.

2.2.2.2 Activity Modeling

The travel-based activities undertaken by individuals in a day can be analyzed [88] with the aid of GPS tracks. Examples include travel to work, travel to a food court for lunch, after-work travel to fetch children, and travel home. Finding out the sequence of activities and destinations is useful for understanding the needs and patterns of pedestrians, and which can be used in a simulation to study the aggregate behavior of pedestrians.

Sorensen [141] studied the thought processes and activities of shoppers and modeled their dynamics in departmental or convenience stores. For example, it was discovered that most shoppers prefer to move clockwise from the entrance of a store to the cashier. The sizes of the aisles and placements of promotion items were also considered. GPS trackers were installed on shopping trolleys so that the movements of shoppers can be tracked.

More recently, there have been some work on modeling pedestrians in train stations and from travel diaries. Stubenschrott et al. [143] modeled how pedestrians in train stations may choose alternative paths if the shorter paths were congested. Alivand et al. [10] modeled how tourists

selected routes based on their travel diaries and scenic photos posted on the Internet. Hanseler [61] learned an origin-destination frequency table for a train station using multiple observation modalities, including travel surveys, tracking of pedestrians in an array of cameras, the train timetable and ridership data.

2.2.3 Analysis of Pedestrian Perception and Cognition

It is of interest to know how pedestrians think and are influenced by surrounding objects. Ohno and Wada [108] look into how pedestrians respond to signboards by conducting experiments on human subjects in virtual reality environments. In [163], it was shown how the bonding effect between pedestrians can have a negative impact on the speed of evacuation. Bonding effect is the opposite of social force, causing pedestrians to move closer to each other.

Decision making in groups [99] is affected by groupthink, group polarization and difficulties in reaching consensus. Surowiecki [145] studied the wisdom of crowds and crowd psychology. For example, stock market behavior may sometimes be characterized by crowds blindly following the actions of other investors. He also suggests that when the knowledge of a crowd is combined wisely, it can lead to better decisions. There are also models of opinions in groups [119], wherein individual opinions are influenced by other people in the group. Some pedestrian simulation agents are able to take on different roles such as being the leader and instructing other agents [120]. Leadership and communication behavior can be used to model more complex evacuation in buildings during fire outbreaks.

2.2.4 Modeling of Crowd Disasters

Crowd disasters are those that are directly due to the presence and behavior of large crowds [78]. They may be caused by crowd instabilities that occur when there are sudden changes in the movement or density in some parts of the crowd. Manifestation of crowd disaster dynamics

include crowd turbulence, panic behavior [64], stop and go waves, and crowd pressure [65] in different parts of the crowd. Crowd disasters include stampedes in which people may be trampled on or get crushed, resulting in injury or suffocation.

2.3 Human Factors in Urban Design

In this section, we will survey space syntax in architecture theory and its relation to human activities and densities. We will also look at the evaluation of urban design.

2.3.1 Space Syntax and Influence on Crowds

Architecture is about connecting spaces in buildings to satisfy human needs [57]. Space syntax is about techniques and theories for analyzing spatial configurations of urban structures [67, 68]. It is not only about maximizing the utilization of spaces but also about enabling occupants to fluently transit between spaces, e.g. through well-designed placement of rooms, common areas, corridors and staircases.

Spaces in cities and buildings can be represented by space syntax [67, 68]. Hillier uses graph theory to model the connections of spaces, and can detect local hubs in a road network using graph centrality. Graph centrality measures how central is a node in relation to to other nodes, and can help identify city centers in a city layout. Isovis and axial space are used frequently by architect to represent space volume and straight line of sight respectively. Isovis [146] represents all the distances a person can view in all directions from a point inside a building. Axial space represents road networks as a minimal set of connecting straight lines, and can be used for identifying long stretches of road. These roads are usually the main roads to which smaller roads connect and are local hubs which are densely surrounded by buildings.

2.3.2 Evaluation of Urban Design

Aschwanden et al. [14] used pedestrian simulation to evaluate an urban layout. O'Sullivan and Morrall [111] analyzed the catchment area of train stations, which is the surrounding area from which potential passengers will use a particular train station instead of another station or alternative means of transport. Chen et al. [30] looked into transportation demand and accessibility. Transportation demand measures the number of people using a particular transportation route, such as a bus route, at a particular time of the day. The transportation capacity must be designed to handle the level of demand. Accessibility is the measure of how long it takes for a person to travel from one location to another. Parker et al. [118] worked on simulating land use in the city. Land use is affected by location, price, and accessibility. Frank et al. [50] measured how walkability in a city indirectly improves the health of residents.

Urban planners and government often use zoning when it comes to city planning. Zoning is the process of segmenting and classifying land into different types for residential, commercial, industrial and agricultural use. The purpose is to prevent incompatible activities from being located in proximity, such as having heavy industry side next to residential areas. Zoning at mall level is useful for shopping mall management to allocate shops based on the functions of the shops (for example eateries can be located near the mall entrance and fitness centers can be located in the remote region of the mall which is much further from the mall entrance) and transfer between shops. Figure 2.1 shows an example of zoning plan in a shopping mall.

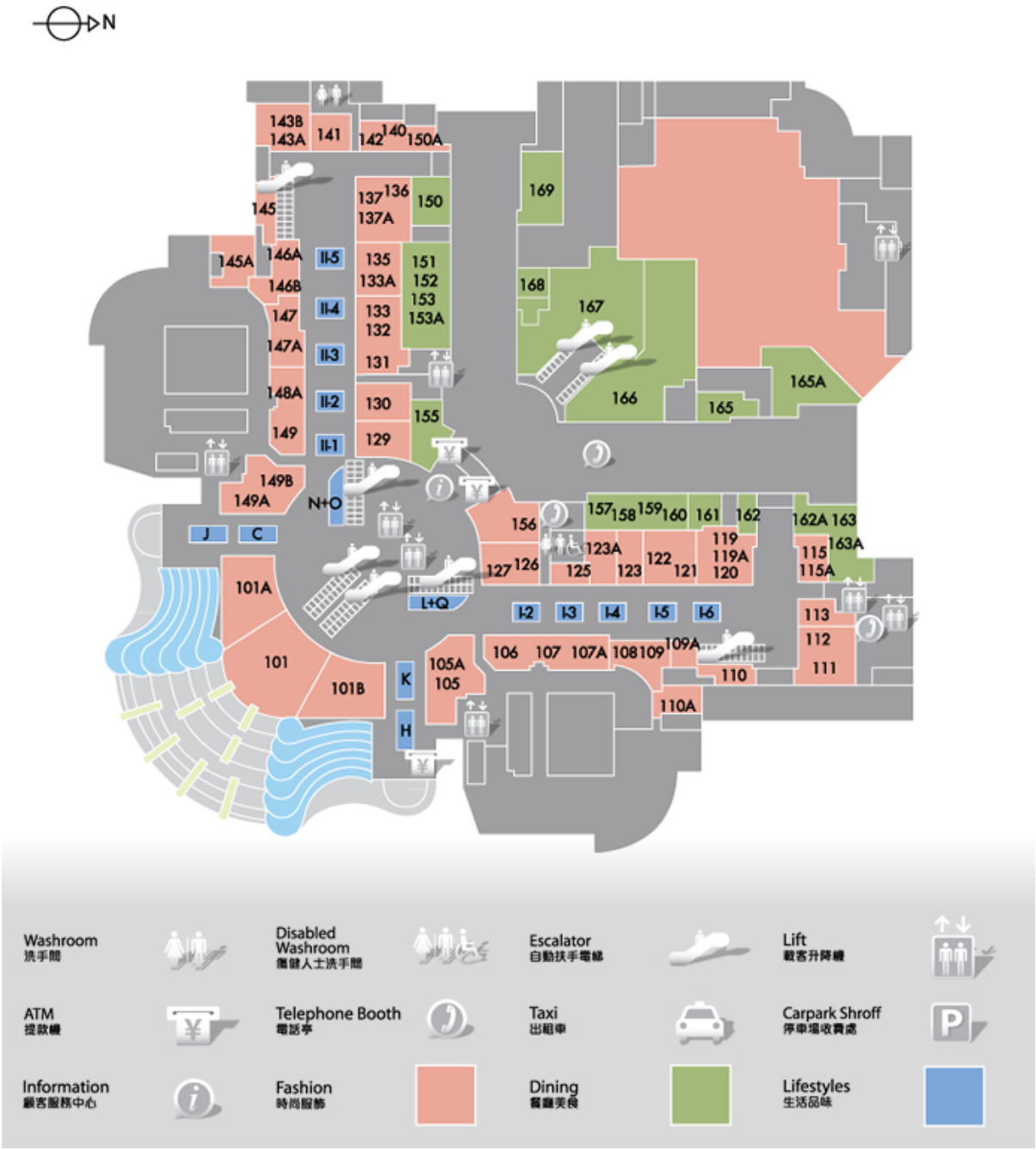


Figure 2.1: An example of a zoning plan. This plan is for the Grand Gateway 66 shopping mall. Image taken from http://www.grandgateway66.com/.

Chapter 3

Pedestrian Detection Using Second Order Information

3.1 Overview

Pedestrian detection in images remains a challenging problem due to the large range of environmental factors influencing the imagery, such as background clutter and lighting variation. Furthermore, there are multiple classes of objects with dominant vertical aspect ratios that masquerade as human shapes when analyzed via conventional image features, resulting in false detection.

A standard approach to pedestrian detection is the use of sliding windows for classifying different image patches, based on a variety of features extracted from each patch. One of the most popular features is the Histogram of Oriented Gradients (HOG [33]). Here a shape is represented by a distribution of local edge directions, which is more robust to variations and noise as compared to a representation that directly uses specific edge positions.

According to [21], HOG and Local Binary Patterns (LBP) are the most commonly used features in single frame pedestrian detectors; as such, we use the HOG-LBP detector [156] as a baseline for analysis and comparison in this thesis. However, HOG and LBP have limitations.

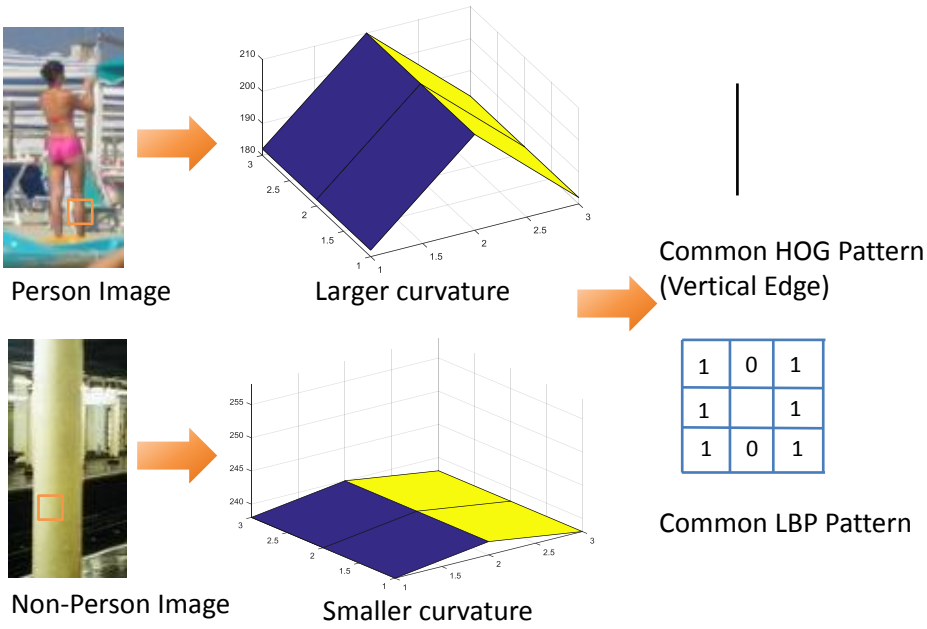


Figure 3.1: Comparing the image intensity surfaces from the person (top) and non-person (bottom) images. Notice that the patches on both person and non-person images have the same HOG and LBP patterns, but they have different intensity curvatures. These curvatures are useful to discriminate between the two images. ©2015 IEEE

Consider the example shown in figure 3.1: we see that both HOG and LBP capture image discontinuities well, and in particular the distribution of edge directions. While they are useful in many instances, they invariably have problems in distinguishing between different objects that have certain shapes and aspect ratios, and these failure cases are intuitively predictable. In figure 3.1, neither HOG nor LBP are able to distinguish between the person and non-person images.

Looking more closely at the image patches themselves, it may be seen that the smooth variations of intensities are in fact different. One approach to capture this additional second order information is via the image intensity Hessian, which captures the principal curvatures of the intensity surfaces.

Second order intensity features have been used in computer vision. Curvature-based regularization has been used for denoising using Euler elastica [16]. Gabor wavelet detects curvature for

image representation [84]. Second order priors were also used for stereo reconstruction [160]. In our case, we only consider intensity patches in a 3×3 neighborhood for reduced computational cost.

In subsequent sections, we explore not just how the use of Hessian features will lead to improved classification, but also *why*. We will not only show more intuitively the manner in which Hessian features discriminate between different image patches when used with a linear Support Vector Machine (SVM), but also how the utility of Hessian features changes when used in conjunction with HOG and LBP features.

We repeat from Chapter 1 the contributions in this chapter:

- We proposed a new feature which captures second order intensity variation. We have shown that when this feature is combined with HOG (Histogram of Oriented Gradients) [33] and LBP (Local Binary Patterns) [156], it leads to more than 10% improvement in pedestrian detection accuracy in most datasets. Use of these features is also more computationally efficient in inference and learning than deep learning approaches. This enables implementation in low power embedded systems and operates in interactive time.

- We presented various novel insights from carrying out in-depth analysis on failure modes of previous features. We highlighted new patterns that our new feature is capable of recognizing by detailed visualization of the responses from our Support Vector Machine (SVM) detector. We also analyzed the weights of corresponding features in the SVM to see how they adapt when our feature is combined with other features.

We introduce a Hessian-based feature which is simpler and computationally less expensive for both training and execution compared to deep learning methods and is thus more appropriate for embedded or lower powered hardware. There is also a comparatively lower risk of overfit-

ting [105] to a particular dataset due to unintended but systematic differences in lighting and backgrounds across different datasets.

In our case, we restrict our focus to the conventional sliding-windows with linear SVM approach, so as to be able to present interesting and intuitive insights into how the different features assist in the detection task.

3.2 The Hessian-HOG-LBP Model

In this section, we first describe our second order Hessian features and then provide details of our implementation. More interestingly, we will shed light on what happens when Hessian features are used in conjunction with HOG and LBP features for training a linear SVM pedestrian classifier. In particular in section 3.3 we will compare how classifier weights for Hessian features are different when they are used alone (whereupon they behave in an analogous manner to HOG and LBP features), as compared to when they are used with HOG and LBP features and can thus adapt and focus on more distinctive image details that are neglected by HOG and LBP.

3.2.1 Hessian and Curvature

The Hessian is a square matrix that contains second partial derivatives of a multi-variable function. For a function of $f(x,y)$, the Hessian is a 2 by 2 matrix of the form

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \tag{3.1}$$

where x and y are the variables of the function. An image may be treated as a function of two variables where the output of the function is the pixel intensity value (see figure 3.2), while x and y are spatial coordinates in the image.

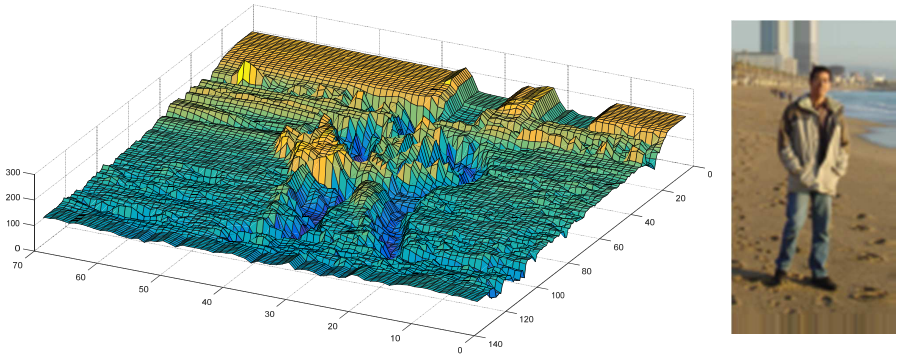


Figure 3.2: Example of an image seen as a function of two variables where the output of the function is the pixel intensity value (left) and the original image (right).

The eigenvectors of the Hessian matrix are the directions of the principal curvatures of a function, with the absolute value of an eigenvalue determining the curvature magnitude of the corresponding principal curvature and the sign determining whether it is convex or concave. The Hessian of an image may be visualized as the shape of the local intensity surface, which takes the following forms: The principal curvatures are not represented jointly as a joint distribu-

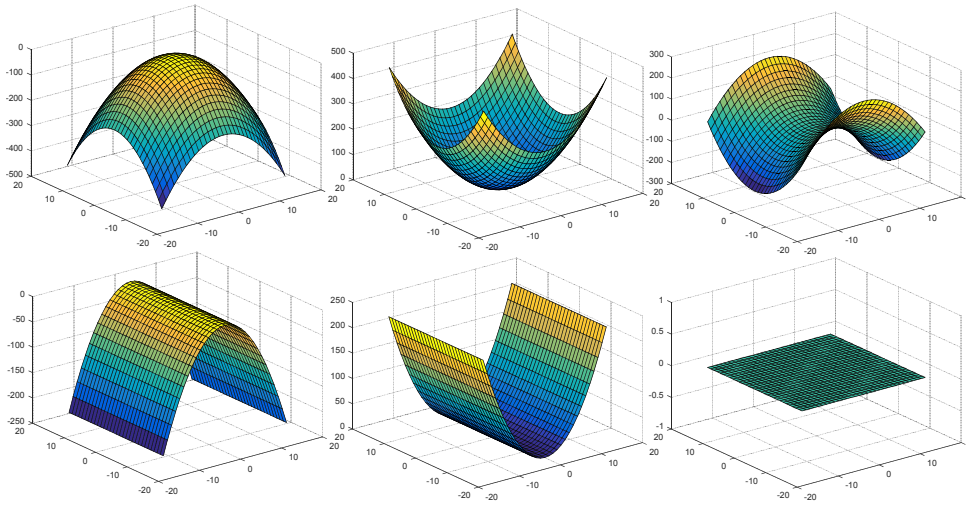


Figure 3.3: Types of curvature found in a small image patch. From top to bottom and left to right: local maximum, local minimum, saddle, ridge, valley and flat surface.

tion histogram but instead each principal curvature (two principal curvatures in total) is voted separately into one of the two histograms depending on whether the curvature is positive or negative. Two principal curvatures multiply by two histograms gives us four types. We encode only four types to make the feature vector length smaller and therefore more practical. We will explain our method in more details in subsequent sections.

In computer vision, the Hessian is used in the Hessian affine detector [95] to detect interest points which are local extremas of both the determinant and trace of the Hessian matrix.

3.2.2 Hessian Features

To understand Hessian features, let us first understand how the HOG feature is computed. The equations in (3.2) are used to calculate magnitudes and angles of edges at every pixel:

$$\begin{aligned} dx_{j,i} &= I_{j,i+1} - I_{j,i-1} \\ dy_{j,i} &= I_{j+1,i} - I_{j-1,i} \\ \text{mag}_{j,i} &= \sqrt{dx_{j,i}^2 + dy_{j,i}^2} \\ \text{ang}_{j,i} &= \arctan(dy_{j,i}, dx_{j,i}), \end{aligned} \tag{3.2}$$

where I refers to the image, while mag and ang are the computed magnitudes and angles of each edge respectively. A histogram is computed for the magnitudes for each cell of 8 by 8 pixels. The histogram is normalized for every adjacent 2 by 2 block of cells, using a clipped L2-norm [33]. The full set of histograms form the HOG descriptor.

To compute the Hessian descriptor in an image of dimension 64×128, we first normalize the image gamma via a square root intensity transfer function. Next, using second order derivatives of image intensities:

$$\begin{aligned} dxy_{j,i} &= I_{j+1,i+1} + I_{j-1,i-1} - I_{j+1,i-1} - I_{j-1,i+1} \\ dxx_{j,i} &= I_{j,i+1} + I_{j,i-1} - 2I_{j,i} \\ dyy_{j,i} &= I_{j+1,i} + I_{j-1,i} - 2I_{j,i}. \end{aligned} \tag{3.3}$$

We can compute the Hessian matrix **H** and its eigendecomposition (computable in closed form):

$$\begin{aligned}\mathbf{H} &= \begin{bmatrix} dxx & dxy \\ dxy & dyy \end{bmatrix} \\ &= \mathbf{W} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \mathbf{W^T} \\ &= \begin{bmatrix} \mathbf{u_1} & \mathbf{u_2} \end{bmatrix} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} \mathbf{u_1}^T \\ \mathbf{u_2}^T \end{bmatrix}.\end{aligned} \tag{3.4}$$

If color images are used, Hessians are separately computed for each RGB color plane and the Hessian with the largest absolute determinant is selected.

The eigenvalues and corresponding eigenvectors are then partitioned into four groups based on whether the eigenvalues are positive or negative, and whether the eigenvalues are dominant or weaker (since eigenvalues come in pairs). Subsequently, histograms are computed for these four groups by weight-voting the eigenvalues and eigenvectors into spatial and orientation cells, similar to HOG [33], except that there are four histograms instead of one. Contrast normalization is also performed within every block across histograms as per [33]. Next, the Hessian descriptor is concatenated with HOG and LBP descriptors. The merged descriptor then represents our full image descriptor, and can be used for training or classification by a linear SVM (see figure 3.4).

An illustration of the different Hessian features extracted from a single image containing a person is shown in figure 3.5.

In HOG [33], a scanning window is 128 pixels in height and 64 pixels in width. The scanning window is divided into 16 by 8 cells. Each cell is a 8 by 8 image patch. Each cell has 9 bins for each orientation. A vote is cast by the gradient of the pixel into four nearest cells and two nearest orientation bins. The size of the vote is the magnitude of the gradient. A block is formed by 2 by 2 adjacent cells. Each block of features is contrast normalized separately. The

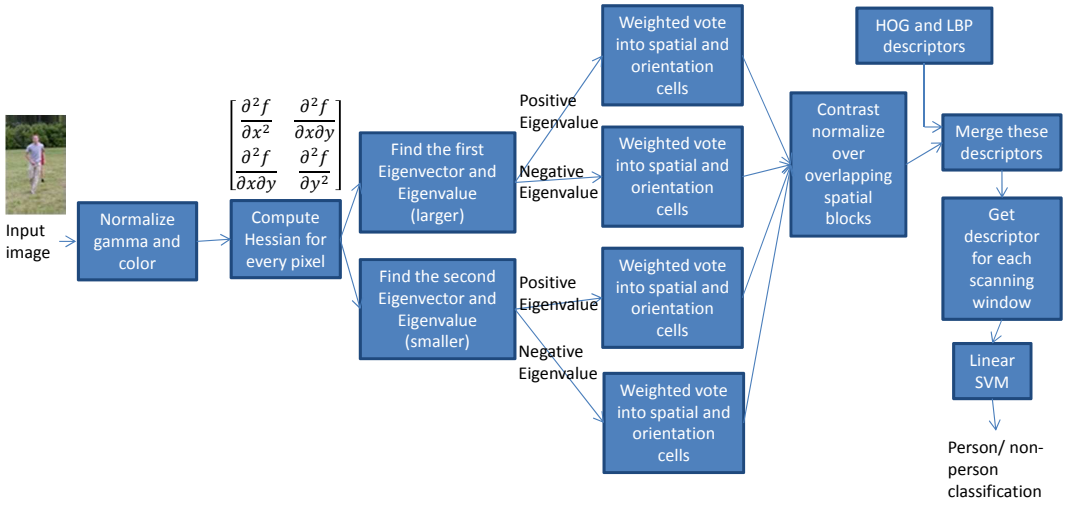


Figure 3.4: Block diagram of our detector.

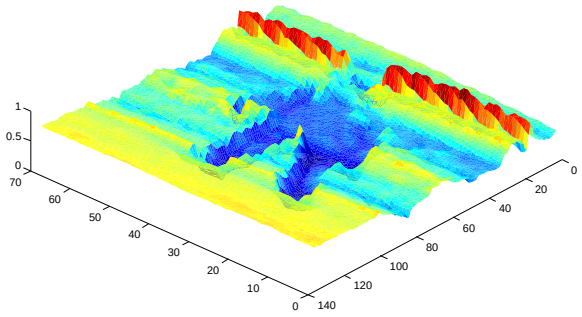


Figure 3.5: 3D intensity surface of an image. ©2015 IEEE

HOG descriptor is made up of 15×7 blocks which have a feature size of $15 \times 7 \times 4 \times 9 = 3780$ dimensions.

There are 3780 and 1888 dimensions in the HOG [33] and LBP [156] respectively. Hessian features are represented in 15 $\times$ 7 blocks $\times$ 2 $\times$ 2 cells $\times$ 2 eigenvectors $\times$ 2 (separate positive/negative eigenvalues) $\times$ 4 directions = 6720 dimensions (note: same number of blocks and cells as conventional HOG). A cell is a non-overlapping 8 by 8 pixel. Every adjacent 2x2 cells is combined to form a block. Eigenvectors are derived from the Hessian matrix with different

histogram bins for eigenvectors with positive and negative eigenvalues. We reduce the number of direction bins for each histogram to 4 instead of the 9 used in HOG, to make the feature size more manageable.

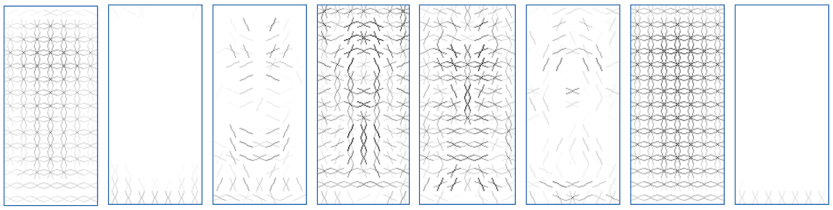
For images larger than 64x128, pedestrian detection is performed by using the sliding windows approach, across overlapping shifts and scales. At each window position, the combined descriptor is extracted and classification is performed.

3.3 Adaptation of Hessian Weights with HOG and LBP

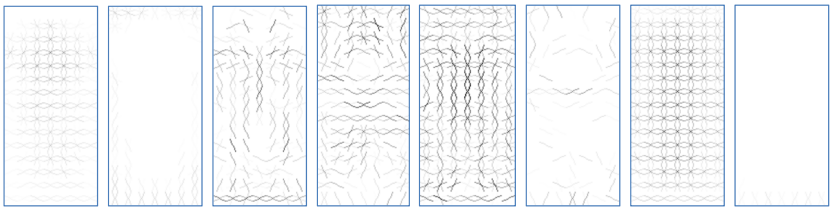
The classifier we use for our detector is a Support Vector Machine (SVM) with a linear kernel. We want to understand how different features and different patches respond in a manner that leads to either a positive or negative classification. To visualize this, we use a method similar to occlusion detection by Wang et al. [156]. First, we want to consider which components of the Hessian descriptor are more significant than others, based on their SVM-related weights. In particular, we are also interested in how these weights change when the Hessian descriptor is used in conjunction with HOG and LBP, as opposed to when it is used alone.

Figure 3.6 shows the different SVM-related weights of the Hessian descriptor, obtained separately when (a) the Hessian descriptor is used in isolation, and (b) when it is used within the combined Hessian-HOG-LBP descriptor.

The Hessian weight patterns are visualized by drawing short edges that are perpendicular to the directions of the curvatures – the darker an edge, the higher its weight. We can visually compare the (a) and (b) images. In particular, consider the fourth and sixth columns in figure 3.6. We can intuit that the Hessian feature tries to detect the contour of a person when used alone (the human shape shows up in this visualization of the weights). However, when the Hessian feature is combined with HOG and LBP, the "task" corresponding to contour detection has



(a) Hessian Only Detector weights



(b) Hessian weights of our fusion Detector

Figure 3.6: Left to right: First pair shows negative and positive weights for Hessian with 1st eigenvalue negative in value, second pair for 1st eigenvalue positive in value, third pair for 2nd eigenvalue negative in value, fourth pair for 2nd eigenvalue positive in value. (a): A detector trained on Hessian features only. (b): A detector trained on Hessian, HOG and LBP features. Some parts of Hessian is used to detect the shape of a human and some parts of Hessian detect curvature and offload the detection of human shape to HOG. ©2015 IEEE

been offloaded to HOG, with the Hessian weights instead adapted to focus more on within-body intensity variations that are less easily captured by HOG and LBP.

Positive SVM weights are patterns found in positive samples in the training set, and likewise negative SVM weights are patterns found in negative samples.

3.3.1 Visualization of SVM Responses

Our SVM is a linear classifier that has one bias. Since there is a bias, we are unable to compare the output that is contributed by a non-overlapping subset of the input vector. To solve this problem, we use the distribution of bias described in the HOG-LBP paper [156].

We distributed the bias of our linear SVM to individual spatial blocks (patches) as well as each feature type of HOG, Hessian and LBP, such that each feature-block combination may be considered an independent SVM with its own bias.

In figure 3.7, the responses of features extracted from each patch are color-coded to show how these influence the SVM classification of the image as containing a person or not — i.e. Red color represents positive values and green color represents negative values. Redder responses skew the classification towards a "person" label while greener responses do the opposite. The numbers below the HOG, Hessian and LBP feature outputs are the scores returned from each feature by the detector, where each score is the sum of the block scores for the corresponding feature. A more positive score means that the corresponding feature is interpreting the image to be more human-like. The score from the Hessian only detector is smaller than the scores from the individual component features of HOG-Hessian-LBP detector, and consequently it is also smaller than the combined score of the HOG-Hessian-LBP detector, obtained by summing the individual component feature scores. While it is obvious that the fused Hessian-HOG-LBP descriptor is better than just using the Hessian descriptor, what is interesting is that the Hessian descriptor when used by itself *behaves like the HOG descriptor* — note the positions of the deeper red and green blocks. However, when used within the combined descriptor, the Hessian component *adapts to picking up other image details not handled by the other components*, in this case is the presence of a floor area with relatively uniform intensity.

3.4 Analytical Comparison of HOG, LBP and Hessian Features

From our experiments (quantitative results are covered in section 3.5.2) using the INRIA dataset [33], many false positives (see figure 3.8) were caused by strong vertical edges (e.g. buildings, lamp posts and other poles) and strongly textured patterns. In many of these false positive cases, the pedestrian image patches had sufficient information to allow for a correct classification, but these were not captured by HOG or LBP features. As we will show in later examples, the encoding of luminosity curvatures via the Hessian can resolve many of these problems.

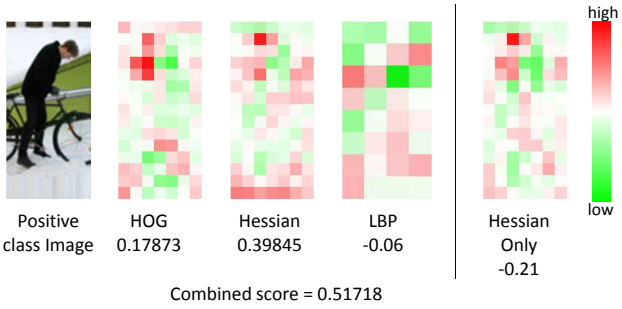


Figure 3.7: Example of comparing the responses from individual components of the combined descriptor as compared to the Hessian-only descriptor detection. Redder colors indicate locations of the image for which the features respond positively while greener colors mean the opposite. The labels "HOG", "Hessian" and "LBP" refer to the different features used within the detector. The numbers below the HOG, Hessian and LBP feature outputs are the scores returned from each feature by the detector, where each score is the sum of the block scores for the corresponding feature. (Needs to be viewed in color.) ©2015 IEEE

Consider a 3×3 image patch around a pixel. With reference to such a patch, figure 3.9 illustrates how LBP encodes information related to the shape type (edge, point or corner) while HOG captures contrast strengths and angles of the edges. Extracting the image patch Hessian will additionally describe the strengths and angles of the principal curvatures, which may qualitatively be thought of as cylinderness, pointness or cornerness of the patch. Using all three types of features give us a richer representation of the patch.

To investigate how different features assist one another and fill up weakness gaps when used together as a joint HOG-Hessian-LBP detector, we compare the weights and detection responses of each feature component within the HOG-Hessian-LBP detector in the sections below.

3.4.1 When Hessian is Better than HOG

In figure 3.10, it can be seen that the HOG feature detects the building as a positive (lots of red) because the building image contains strong vertical lines. Conversely, the Hessian descriptor tends to reject the person label (lots of green) because of strong curvatures in the

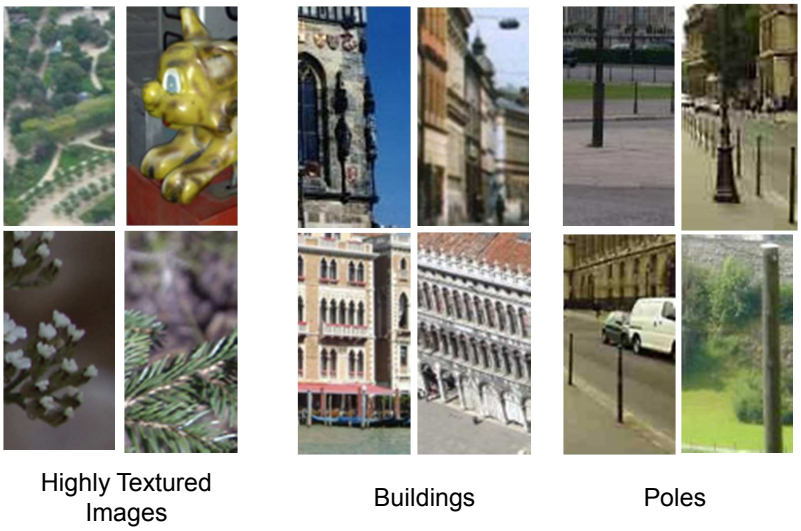


Figure 3.8: Three usual types of false positives, involving highly textured images, buildings and poles.

building image. The numbers below the images are the scores returned from the detector. Note that although the LBP descriptor on average responds negatively to this image, the response is weak and will be dominated by the HOG descriptor. In contrast, the Hessian descriptor's negative response is strong and helps classify the image as "non-person".

3.4.2 When Hessian is Better than LBP

In figure 3.11, the upper half of the image is detected by the LBP feature as positive. This is due to LBP patterns (shown in the figure) commonly found in the upper part of the human body being also present in the upper part of this image. Conversely, the Hessian descriptor returns negative-class responses based on image patterns found in the image. The Hessian descriptor will help reject this image as containing a person.

The effective Hessian feature responses shown in figures 3.10 and 3.11 fit in with the observations made in section 3.3 and from figure 3.6, that the second order information extracted by

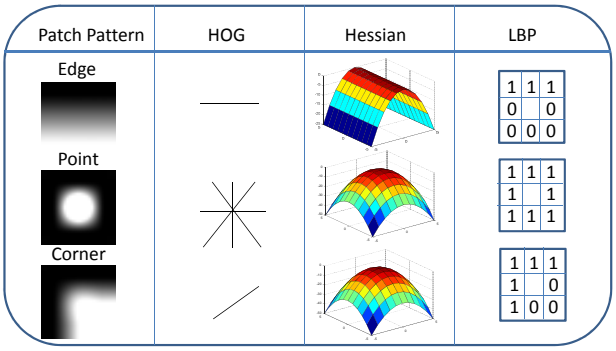


Figure 3.9: HOG, Hessian and LBP encode different properties of a 3×3 image patch. ©2015 IEEE

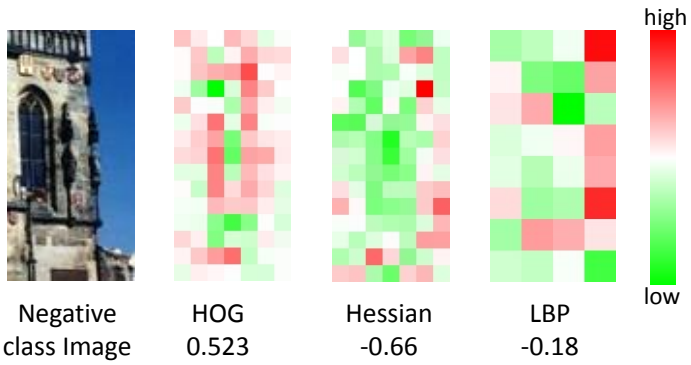


Figure 3.10: Example of a building image for which the HOG feature within our fusion detector responds false positively while the Hessian feature responds negatively. ©2015 IEEE

the Hessian feature is complementary to HOG and LBP, and is thus useful in recovering from the failure modes of both HOG and LBP.

3.4.3 Detection Support from Different Parts of the Hessian Weights

There are four histograms in the Hessian descriptor, which we will analyze individually. Each of these histograms appears to capture different important feature patterns that are important for accurate pedestrian classification.

In the subsequent sections, the eigenvalues are ordered in signed magnitude. Hence, the "first"

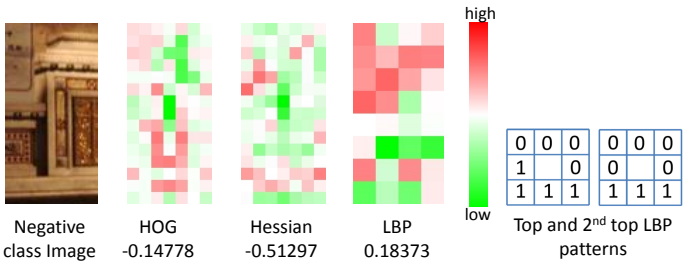
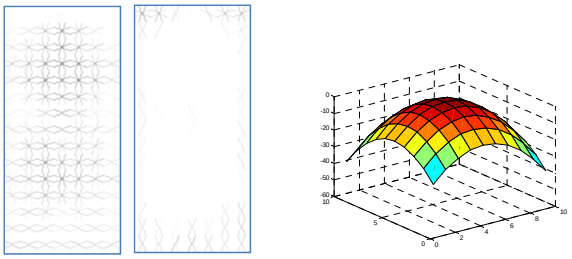


Figure 3.11: Example of a non-person image for which the LBP feature responds false positively while the Hessian feature responds true negatively. For reference, the two 3×3 arrays shown are the most common LBP patterns found in pedestrian images. ©2015 IEEE

eigenvalue represents the most positive eigenvalue, even if in absolute magnitude it is smaller than the other eigenvalue. The eigenvector having a larger (signed) eigenvalue will be contributed to the histogram of the first eigenvector. Likewise the eigenvector having a smaller (signed) eigenvalue will be contributed to the histogram of the second eigenvector.

3.4.3.1 Hessian Weights for Negative First Eigenvalue Histograms

Since the first eigenvalue (which is larger than the second eigenvalue) is negative, the second eigenvalue must be negative. It detects image surface with lots of local maxima as background (see figure 3.12).



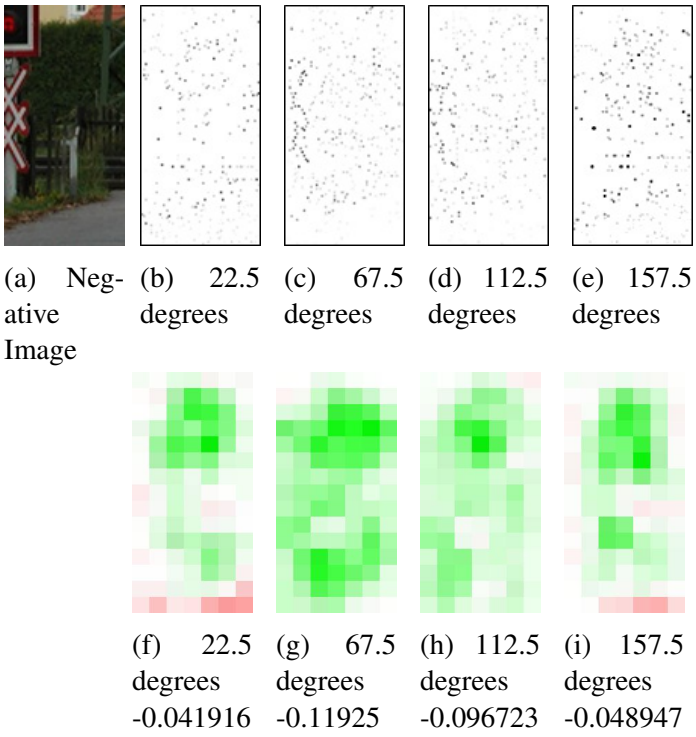
(a) Negative and positive SVM weights

(b) Shape of the image surface

Figure 3.12: (a):SVM weights when the Hessian first eigenvalue is negative. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect.

From figures 3.13 and 3.14, we can see that the negative image has an image surface that has

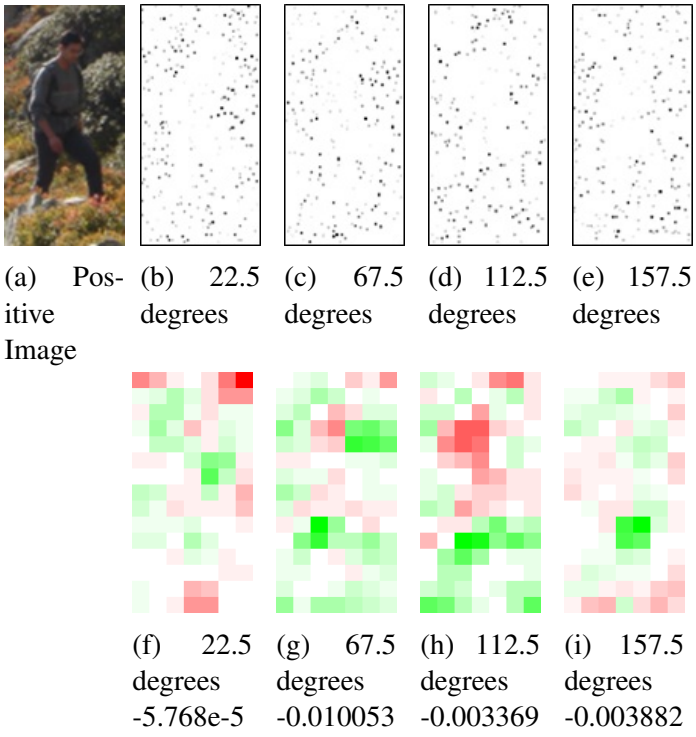
lots of local maxima. The positive image has fewer local maxima in the center of the image (the pedestrian region) and lots of local maxima in the surrounding part of the image (background region). The overall output of this part of the Hessian weights of the positive image is higher than the negative image. This helps to differentiate background images from pedestrian images.



(a) Negative Image (b) 22.5 degrees (c) 67.5 degrees (d) 112.5 degrees (e) 157.5 degrees

(f) 22.5 degrees -0.041916 (g) 67.5 degrees -0.11925 (h) 112.5 degrees -0.096723 (i) 157.5 degrees -0.048947

Figure 3.13: (b) to (e): Locations and magnitudes of the 1st eigenvalue which is negative for different orientation of the 1st eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 1st eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed [156] to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.)



(a) Positive Image (b) 22.5 degrees (c) 67.5 degrees (d) 112.5 degrees (e) 157.5 degrees

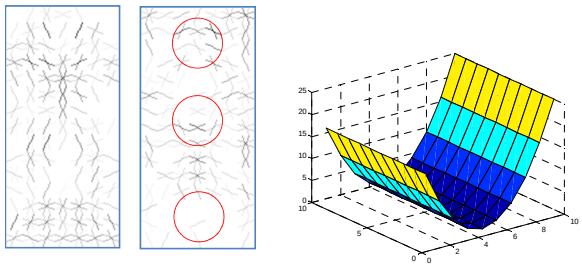
(f) 22.5 degrees -5.768e-5 (g) 67.5 degrees -0.010053 (h) 112.5 degrees -0.003369 (i) 157.5 degrees -0.003882

Figure 3.14: (b) to (e): Locations and magnitudes of the 1st eigenvalue which is negative for different orientation of the 1st eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 1st eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.)

3.4.3.2 Hessian Weights with Positive First Eigenvalue Histograms

From figures 3.15 and 3.16, we can see some similarities and differences between their weight patterns. Both weights detect the contour at the top of the head and shoulders and reject images with strong vertical lines in the center of the torso. The two weights differ at the center of the head, the upper part of the pants region and the feet region as highlighted by the red circles drawn in the images. HOG weights are useful for detecting the shape of the head and top part of the pants (lots of vertical lines). Conversely, it appears that these parts of the Hessian weights are targeted for detecting the textures of the head and the upper part of the pants (by detecting

horizontal ridges). In the feet region, the HOG feature favors horizontal edges for the bottom of the feet and vertical edges for the pants, while the Hessian feature prefers to reject a heavily textured ground. To visualize the positions of these regions, look at figure 3.17 in which the corresponding regions are also circled in red.



(a) Negative and positive SVM weights

(b) Shape of the image surface

Figure 3.15: (a):SVM weights when the Hessian first eigenvalue is positive. The darker edge implies larger weight. The red circles highlight the regions where the Hessian and HOG weights are the most different (see figure 3.16). (b): Shape of the image surface that these weights try to detect.

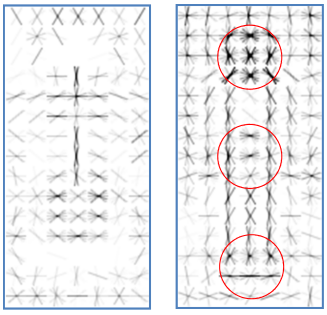


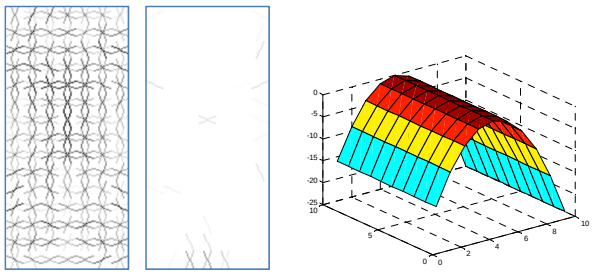
Figure 3.16: Negative and positive SVM weights for HOG. The darker edge implies larger weight. Note that the edges have nine orientations. The red circles highlight the regions where the Hessian and HOG weights are the most different (see figure 3.15).

Figure 3.17: The corresponding regions (in red circles) of the HOG and Hessian weights are also highlighted in this example image.

In this section, we limit our analysis to the obvious differences in weights of the Hessian and HOG features, without further analyzing the detection responses as in the previous section. This is because we were unable to find an example image in which the individual component responses of these three regions obviously contribute to a correct detection, with correct detections typically resulting from a combination of different factors.

3.4.3.3 Hessian Weights for Negative Second Eigenvalue Histograms

From figure 3.18, we can see that this part of the Hessian weights is useful for classifying images that are strong in concave intensity curvatures (dark-bright-dark patterns) as background. These weights reject horizontal ridges (strong vertical curvatures) at the legs region as atypical of a pedestrian image. This makes sense as the image regions of legs have strong vertical ridges rather than horizontal ridges.

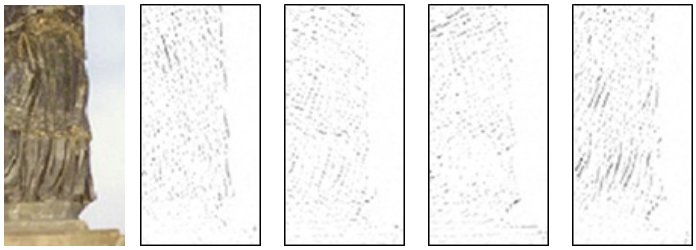


(a) Negative and positive SVM weights

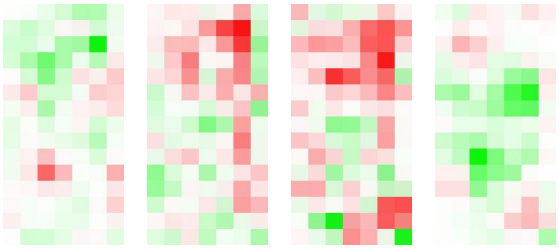
(b) Shape of the image surface

Figure 3.18: (a):SVM weights when the Hessian second eigenvalue is negative. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect.

From figure 3.19, we see that the direction of the second eigenvector that has the largest negative output value depends on the direction of the texture in the image. This helps the detector reject this negative image.



(a) Negative Image (b) 22.5 degrees (c) 67.5 degrees (d) 112.5 degrees (e) 157.5 degrees

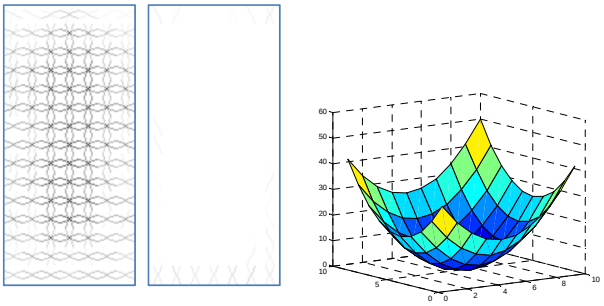


(f) 22.5 degrees -0.1627 (g) 67.5 degrees 0.067063 (h) 112.5 degrees 0.1545 (i) 157.5 degrees -0.33

Figure 3.19: (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is negative for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue negative in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.)

3.4.3.4 Hessian Weights for Positive Second Eigenvalue Histograms

Since the second eigenvalue (which is smaller than the first eigenvalue) is positive, the first eigenvalue must also be positive. It thus detects an intensity surface with lots of local minima as background (see figure 3.20).

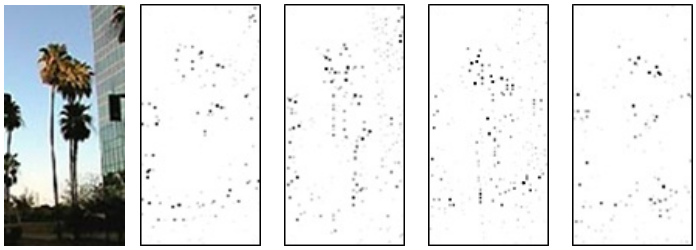


(a) Negative and positive SVM weights

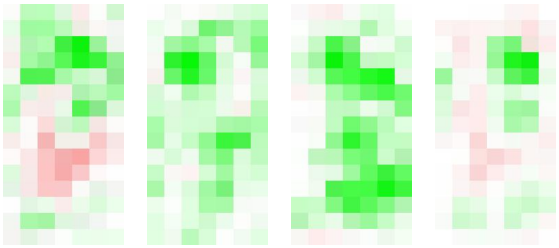
(b) Shape of the image surface

Figure 3.20: (a):SVM weights when the Hessian second eigenvalue is positive. The darker edge implies larger weight. (b): Shape of the image surface that these weights try to detect.

From figures 3.21 and 3.22, we can see that the negative image has an image surface that has lots of local minima. The positive image has fewer local minima in the center of the image (the pedestrian region) and lots of local minima in the surrounding part of the image (background region). The overall output of this part of the Hessian weights of the positive image is higher than the negative image. This helps it to differentiate background from pedestrian images.

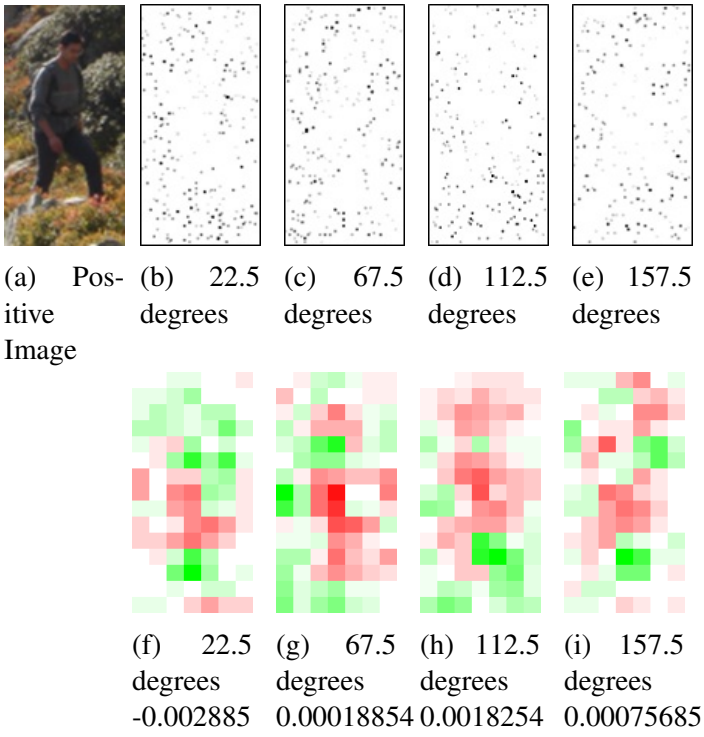


(a) Negative Image (b) 22.5 degrees (c) 67.5 degrees (d) 112.5 degrees (e) 157.5 degrees



(f) 22.5 degrees -0.033482 (g) 67.5 degrees -0.14428 (h) 112.5 degrees -0.13375 (i) 157.5 degrees -0.033103

Figure 3.21: (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is positive for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue positive in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.)



(a) Positive Image (b) 22.5 degrees (c) 67.5 degrees (d) 112.5 degrees (e) 157.5 degrees

(f) 22.5 degrees -0.002885 (g) 67.5 degrees 0.00018854 (h) 112.5 degrees 0.0018254 (i) 157.5 degrees 0.00075685

Figure 3.22: (b) to (e): Locations and magnitudes of the 2nd eigenvalue which is positive for different orientation of the 2nd eigenvector. Note that the larger the magnitude, the darker the image. (f) to (i): Output per block for different orientation for the part of the Hessian descriptor which has 2nd eigenvalue positive in value. Red represents positive values and green represents negative values. Bias is distributed to each block. Score for the particular orientation is shown below the images. All images are contrast stretched to bring out the details. (Best viewed in color.)

3.5 Detection Experiments

Having trained our detector on the INRIA dataset [33], we additionally tested the detector on the following datasets: TUD-Brussel [159], ETH video sequence BAHNHOF [44], Penn-Fudan [154] and DaimlerChrysler [100], to determine if it is robust to different lighting conditions and scenes (car views, street views, walking video (run detection on each frame independently), and black and white images) without further re-training. We evaluated our method using per window and per image measures similar to previous works.

Because the number of possible negative training samples is very large (due to all possible scales and shifts of negative images are also negative samples, unlike positive samples where the pedestrian has to be appropriately sized and centered in the image), it is impractical to use all negative samples for training. So we used bootstrapping where we randomly sample 10 negative samples from each negative image to train the initial pedestrian classifier. Next, we run the initial classifier on the negative images to extract hard negative examples from the negative images and re-train the classifier using both the initial negative samples and hard negative samples to get the final pedestrian classifier. In this way, we avoid using all possible negative samples which makes it practical to train on a desktop machine.

The sizes of the bounding boxes in the datasets are changed to match the pedestrian window sizes in the INRIA dataset. We also separately considered pedestrians that were larger than 60 pixels in height in the TUD-Brussel dataset and 128 pixels in height for the ETH dataset, as most of the pedestrians in the TUD-Brussel dataset were significantly smaller.

3.5.1 Description of Datasets

From table 3.1, we can see that each dataset has its own challenges. The INRIA dataset is the most suitable for training as it has many properly cropped pedestrian images with color and high resolution (128x64). Each dataset contains different views and background (street view vs car view) which is suitable to validate the pedestrian detector based on different applications the detector is implemented for. Color in the dataset will help to improve the performance of our detector as our features use color information.

3.5.2 Numerical Results

In subsequent sections, we can see that our method is substantially better than HOG-LBP. The Hessian features were helpful in localization, leading to improved performance in terms of

Table 3.1: Description of each dataset, highlighting their differences and challenges.

	Number of images (train)	Number of images (test)	Image description	Video or separate images?	Color?	Stereo?	Other challenges
INRIA	288 positive and 453 negative images	614 positive and 1218 negative images	Leisure trip photos	Separate Images	Yes	No	-
TUD-Brussel	-	1793 images	Car view and street view	Both	Yes	Yes	-
ETH video sequence BAHNHOF	-	500 images	Street view from a mobile platform	Video	Yes	Yes	small crop size compared to INRIA
Penn-Fudan	-	170 images	Street view	Separate Images	Yes	No	Cropping width not proportional to cropping height
Daimler Chryler	-	24000 pedestrian images, 25000 non-pedestrian images	Cropped people and non-people images	Separate Images	No	No	Small crop size, different aspect ratio compared to INRIA

false positives per image. Hessian-HOG-LBP greatly reduces the failure cases of HOG-LBP due to the false detections of vertical objects and textured images.

3.5.2.1 TUD-Brussel Dataset

Figure 3.23 shows some sample images from this dataset. Figure 3.24 shows the performance of our Hessian-HOG-LBP detector as compared to the commonly used HOG-LBP detector on the TUD-Brussel dataset. The error curve shows the miss rate (ratio of pedestrian that are not detected out of all pedestrians) against false positives per image (ratio of non-pedestrian detected as pedestrian and total number of images. Note that all detections were carried out using sliding windows and were combined using non-maximum suppression. A detection window is considered detected if it overlaps with the ground truth window by more than 50%). This is a difficult dataset as the miss rate is 0.75 at 0.1 false positives per image. There is 18% increase in accuracy at 0.1 false positives per image using this dataset. It uses the video frames and still images. However we do not use motion information between video frames.

(a) Car view (b) Street view (negative) (c) Street view (positive)

Figure 3.23: Example images from the TUD-Brussel dataset.

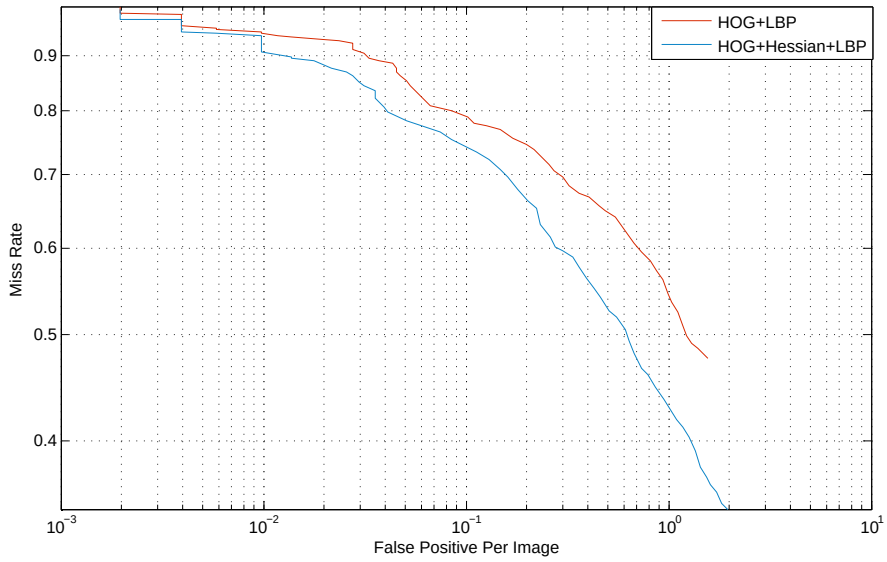


Figure 3.24: Detection error tradeoff curve on the TUD-Brussel dataset. ©2015 IEEE

3.5.2.2 ETH Sequence BAHNHOF Dataset

Figure 3.25 shows one sample image from this dataset. Figure 3.26 shows the performance of our Hessian-HOG-LBP detector as compared to the commonly used HOG-LBP detector on the ETH sequence BAHNHOF dataset. There is 15% increase in accuracy at 0.1 false positives per image using the ETH sequence BAHNHOF.

Figure 3.25: Example image from the ETH sequence BAHNHOF dataset.

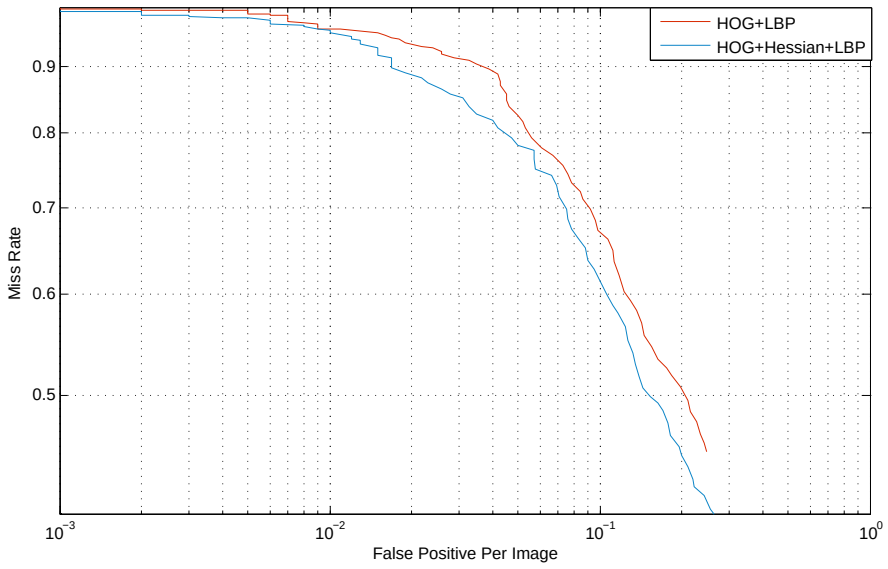


Figure 3.26: Detection error tradeoff curve on the ETH sequence BAHNHOF dataset. ©2015 IEEE

3.5.2.3 Penn-Fudan Dataset

Figure 3.27 shows one sample image from this dataset. Figure 3.28 shows the performance of our Hessian-HOG-LBP detector as compared to the commonly used HOG-LBP detector on the Penn-Fudan dataset. At the standard false positive rate of 0.1 as used in the previous two datasets, our Hessian-HOG-LBP detector has a slightly poorer miss rate of 0.81 compared to the HOG-LBP detector with a miss rate of 0.787. However, the result curves show that our detector performed consistently better in the regime with smaller false positive rates but also smaller detection rates (left half of the graph), with the cross-over point at a false positive rate of 0.075. In this regime, the improvement in detection rate of our detector as compared to the HOG-LBP detector is also substantially higher than the reduction in performance for the regime with higher false positive rates, with a maximum gain of 3.395% in detection rate as a false positive rate of 0.04, compared to a maximum reduction of 1.748% in detection rate at a false positive rate of 0.09.

Figure 3.27: Example image from the Penn-Fudan dataset.

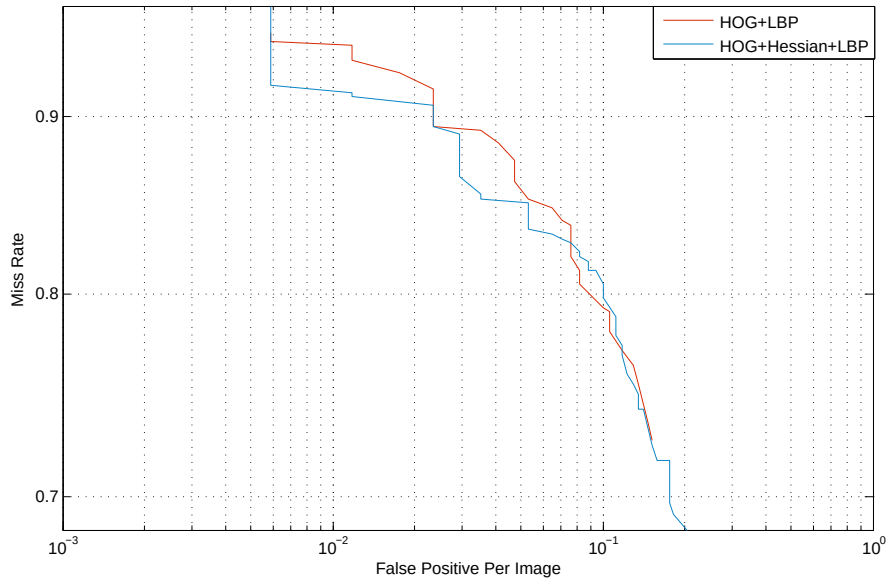
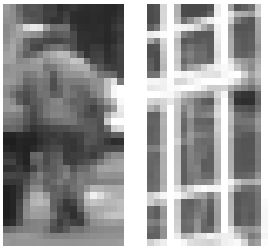


Figure 3.28: Detection error tradeoff curve on the Penn-Fudan dataset. ©2015 IEEE

3.5.2.4 DaimlerChrysler Dataset

Figure 3.29 shows some example images from this dataset. Figure 3.30 shows the performance of our Hessian-HOG-LBP detector as compared to the commonly used HOG-LBP detector on the DaimlerChrysler dataset. Note that this dataset is evaluated using detection rate (ratio of cropped pedestrian images correctly identified as pedestrian) and false positive rate (ratio of cropped non-pedestrian images incorrectly classified as pedestrian). All images in this dataset have a fixed image size and either contains one pedestrian or none and therefore each image is treated as one single detection window. The pedestrian window in this dataset has a different aspect ratio to that in the INRIA dataset on which our Hessian-HOG-LBP detector is trained. This causes the performance of the detector to decrease, which can be improved by training on the DaimlerChrysler dataset instead.



(a) Positive image (b) Negative Image

Figure 3.29: Example images from the DaimlerChrysler dataset.

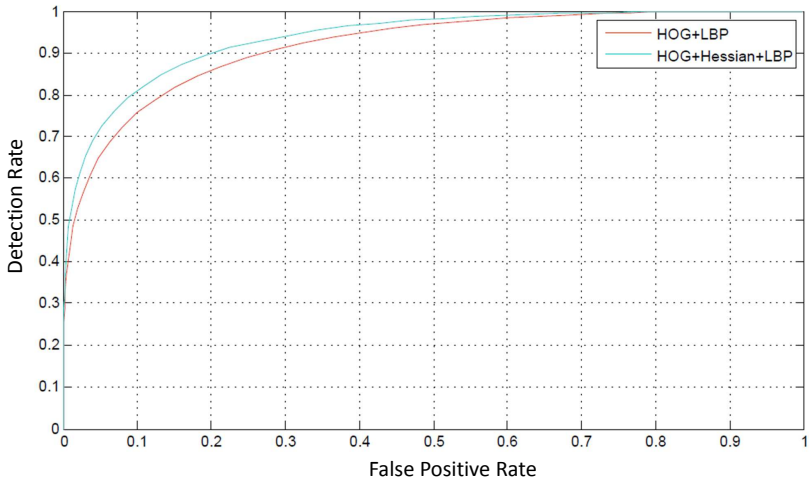


Figure 3.30: Detection rate – false positive rate curve on the DaimlerChrysler dataset. ©2015 IEEE

3.5.2.5 INRIA Dataset

Figure 3.31 shows some example images from the INRIA dataset. Figure 3.32 shows the performance of our Hessian-HOG-LBP detector as compared to the commonly used HOG-LBP detector on this dataset. Interestingly, while there is little improvement in the per window performance on the INRIA dataset, there are a substantial improvements in the per image performance on the other four datasets. This is perhaps because HOG-LBP is better tuned to

the INRIA dataset, but somehow not so well adapted to the other datasets. Note that this dataset is evaluated in false positive per window (ratio of detection windows containing non-pedestrian detected as pedestrian out of all detection windows containing non-pedestrian. Each positive image contains one pedestrian such that each pedestrian is the same size as the image and therefore each positive image is treated as a detection window. For negative images, detection windows from each possible scales and shifts are used).

(a) Positive image

(b) Negative image

Figure 3.31: Example images from the INRIA dataset.

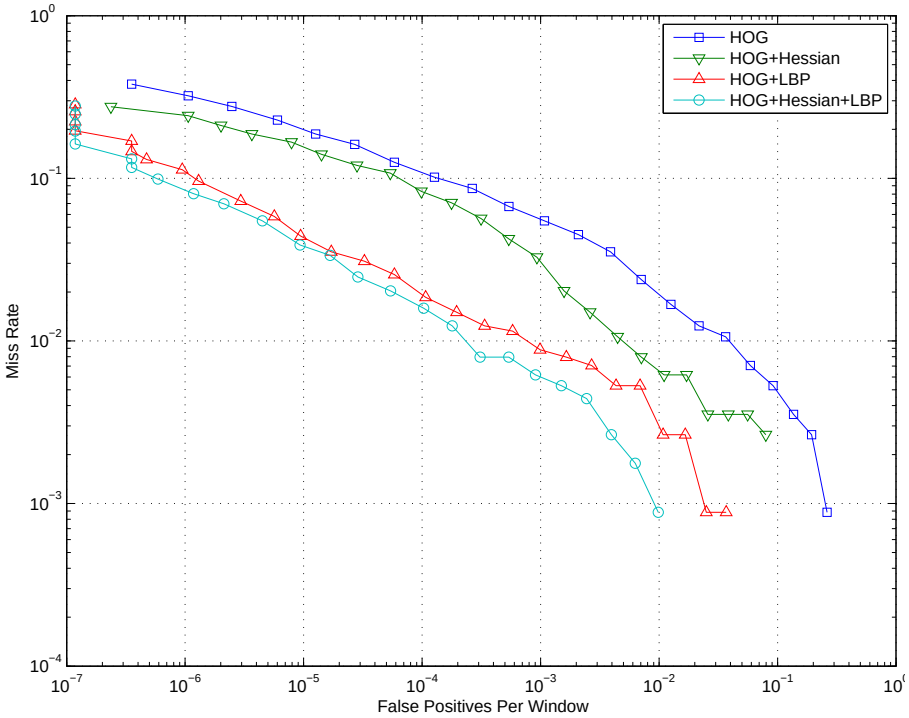


Figure 3.32: Detection error tradeoff curve on the INRIA dataset. ©2015 IEEE

Figure 3.33 shows that our detector is able to find the girl cropped in red while the previous method of HOG+LBP cannot. This is due to the curvature information found in the coat and the legs of the girl providing an extra hint to our detector that leads to better performance.

Figure 3.33: An example where our detector is able to detect a pedestrian missed by the existing HOG+LBP detector.

3.6 Summary

We described and discussed how second order intensity information is useful for overcoming the failure modes of current detectors using our method of visualizing the locations and feature type responses of linear SVM. We have shown that Hessian weight pattern is able to adapt to the new pattern when unified with HOG and LBP. Experimental results demonstrated greater than 10% improvement on three datasets.

Chapter 4

Pedestrian Density Distribution Model

4.1 Overview

The objective of this chapter is to develop a probabilistic model for predicting pedestrian density distributions within a given building layout (see figure 4.1), with a focus on shopping malls. The central idea is to create a model that relates the pedestrian density distribution to known or design parameters of the shopping mall. However as the pedestrian density distribution may vary throughout the day, we make a simplifying assumption that within some mid-length timescales, the distribution is stationary — hence the overall distribution is quasi-stationary. The quasi-stationary assumption is further discussed in section 4.1.1.

Pedestrian simulation methods are often hand-tuned and seldom learned from ground truth data. Agent-based methods [48] model individual agents separately which is computationally very expensive. Conversely, we seek an approach that is fast, by avoiding simulation of individuals, the trade off being that we forego the capability for detailed time-based animations. In addition, some agent-based frameworks treat agents as reactive only to environment factors of the immediate neighborhoods [48], which would be the case if the agents are only in an exploratory mode, or dealing with unexpected emergency crises. This local-only behavior also applies to fluid-based pedestrian simulation [62]. In contrast, we are more focused on domains

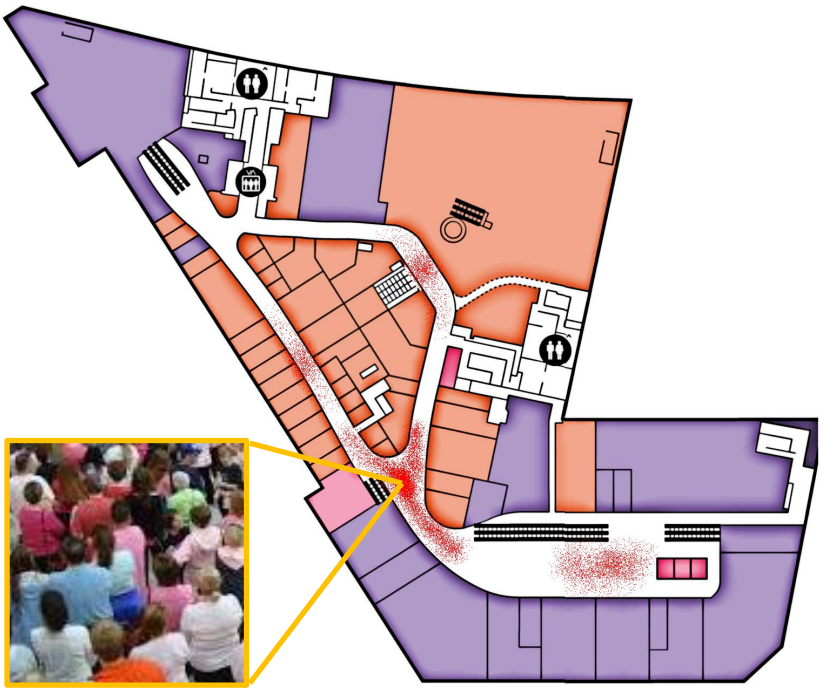


Figure 4.1: Inferring pedestrian density (number of people) distribution in a building layout. Red spots are the possible regions that crowds gather. Floor layout taken from ION Orchard app.

in which the pedestrians have prior knowledge of the entire floor layout, and are doing route planning (perhaps subconsciously) to a larger spatial extent.

Pedestrian density distribution modeling can be useful for shopping mall design. The distribution of shoppers within a mall has an impact on the quality of experience for the shoppers as well as the visibility of different shops. It also has consequences on structural loading and crowd safety.

This distribution depends on the layout of the mall and placement of different shops, and is also affected by the sizes of the mall entrances, the widths of walkways, the locations of escalators and the walking spaces in rooms. In our future vision, the architect creates an initial draft of a floor layout in a shopping mall and provides a tentative shop category zoning plan. Our future system will then automatically and instantaneously generate a predicted density distribution of

shoppers. The architect may then interactively change the design and zoning, and see it affects the shopper distribution.

In this chapter, we will propose a pedestrian density distribution model for which the parameters will be learned from training data. In particular, we will limit the training data to sample pedestrian counts in existing buildings, as opposed to more conventional trajectory data used in previous work, obtained from GPS [88], tracking individuals in camera arrays [7] or Light Detection and Ranging (LIDAR) sensor arrays [27]. Using density data has an advantage of its much greater availability, with sources such as Google traffic, Geographic Information Systems (GIS), geotagged posts on social media and gate sensor data. Collection of GPS data requires greater instrumentation and active participation of users, with significant accompanying issues of scalability and privacy. Depending on camera arrays for collecting extended trajectories is challenging as tracking people in crowded places is still a difficult problem.

4.1.1 Quasi-Stationarity of Pedestrian Density Distribution

People do not remain still, so whenever the notion of pedestrian distribution is discussed, there are the aspects of spatial and time scales. Since we plan to model the distribution non-parametrically in the form of histograms, these scales relate to the sizes of the histogram bins along the spatial and temporal axes.

The spatial scale used in our framework will be described in more detail in section 4.2, but briefly the walking space on a floor is divided into a set of relatively compact nodes.

The timescale, on the other hand, is a more complicated matter. If we were to consider pedestrian distributions at a fine time granularity, say in the order of seconds, the overall distribution is very complex and needs to vary from one time slice to the next based on the motion dynamics of individuals and groups of people. This sort of timescale may be appropriate for tracking applications when the observations obtained are very current, or for intuitive visualization of

hypothetical scenarios, but it will not be possible to predict at this scale what will truly happen on the ground at only the floor plan design phase.

Conversely at the other end of the timescale spectrum, say with a temporal bin size of 24 hours, the distribution may be probabilistically accurate but not useful, since for half the time the mall will be almost entirely empty — so the distribution may be became too diluted for utility.

Instead, we work on a timescale that we think makes the pedestrian distribution *quasi-stationary* temporally. What we mean by this is that the distribution does not change, or only changes minimally, from one time slice (i.e. temporal bin) to the next adjacent slice. We roughly estimate the thickness of each time slice to be on the order of one hour. The distribution will change over multiple time slices, for example from a mall operating hour to one in which the shops are already closed, or from a peak hour to a non-peak hour — hence the term ”quasi-stationary” is used.

In our case, we have chosen to only model the pedestrian density distribution for a non-peak mall operating hour. In addition, we will only model relative densities, rather than absolute densities. This means we will try and predict how much more crowded one specific area of the mall is compared to another, but not the actual number of people. Although pedestrian distributions may depend on absolute densities especially in regions with high congestion, we think this is a reasonable tradeoff as the total population of shoppers in a shopping mall that is still being designed will be impossible to predict and will depend on many external factors, such as the location within a city. Furthermore, using relative densities may mean that our model can apply to a peak-hour distribution, which may only differ from a non-peak hour distribution by a constant factor.

Unfortunately due to resource constraints, we are unable to acquire the ground truth pedestrian distributions in malls by observing entire floors continually for an hour or more. As will be described in further detail in section 4.7.1, our ground truth data consists only of pedestrian

counts in a very short time window, which we will treat as samples from an underlying but hidden distribution.

4.1.2 Contributions

We repeat from Chapter 1 the contributions in this chapter:

- We proposed a highly novel probabilistic model for predicting the approximately steady-state pedestrian density distribution in a shopping mall. The model establishes the Markovian relationship between different latent variables and parameters.

- To our best of knowledge, this is the first time route choice modeling is attempted based on pedestrian count data instead of trajectory histories from tracking instruments such as GPS. The preferences of different routes are hypothesized to be dependent on a number of path descriptors inspired by space syntax theory [67] in architecture, and the route choice model is designed to learn trade offs between these path descriptors. We showed that our model was able to achieve practical accuracy on a shopping mall dataset, fitting the ground truth pedestrian counts better than the baseline objective prior of a uniform multinomial distributions and was also significantly more accurate and faster than a baseline system using agent-based simulation [48].

- We further proposed modeling the shop popularities as latent variables that affect the pedestrian density distributions, but which are themselves influenced by category-specific popularity parameters. We showed that despite learning only from ground truth pedestrian counts, the inferred latent shop popularities correlated well with the measured flow rates into and out of shops.

4.2 Modeling Framework, Assumptions and Conjectures

In our framework, a floor in a mall is modeled as a connected set of nodes. Each **node** represents a non-overlapping region of the *common space outside* shops and other room-level facilities (e.g. toilets), with the common space primarily being the shared corridors that allow a pedestrian to walk from shop to shop. Two nodes are connected if a pedestrian is able to walk from the region represented by one node to the other without traversing any other region. Note that both the topological and geometrical layout of the nodes are important, since in our framework we will consider factors such as line of sight and expanse for modeling pedestrian route choices.

The spaces inside shops are not directly modeled in this framework. Instead the entrances of shops are represented as **portals** from which pedestrians may emerge or disappear. These portals are linked to the appropriate floor nodes depending on the physical locations of the entrances. The shops themselves are formally represented as **vessels**, and may be thought of as "sources" and "sinks" of pedestrians. Note that each vessel may be connected to one or more portals, e.g. some shops are large and have multiple entrances. Besides shops, the other vessels represented are escalators and mall entrances to the building. While escalator vessels are treated as having only one portal each, some mall entrances are so large that they may be modeled as having multiple portals. Formally, a **path** is a sequence of connected nodes that links a portal of a source vessel to a portal of a sink vessel.

One key question that we want to address is this: *at a particular time instant, if we were to randomly select a pedestrian from the entire pedestrian population within the common space, what is the probability that the pedestrian will be found in a particular node?*

There are some simplifying assumptions that we have to make. One, mentioned previously, is the assumption we are modeling for a particular timescale for which the pedestrian density distribution is stationary. Among other things, this means that the pedestrian population remains

constant, and thus the sum of pedestrians leaving through all portals must be equal to the sum of pedestrians emerging from all portals. Second, we assume that each pedestrian is traversing from one portal to another, and ignore cases in which pedestrians are engaged in other activities within the common space, or simply loitering.

Given this framework, a number of other relevant questions arise. When randomly choosing a pedestrian, what is the likelihood that she is traversing from a particular vessel to another? How does this depend on the popularities of the shops and the categories of the shops? Does it further depend on the areas of the shops? What path would the pedestrians choose when walking from one portal to the other, and what do the choices depend on? To address these questions, we rely on some observations and considerations.

First, we have to more formally define the meaning of the term "popularity". Although this will be further elaborated on in section 5.2, we now summarily state that our definition of popularity of a shop is the probability that if a random pedestrian among those transiting between shops (inclusive of escalators and mall entrances) was picked at some random time instance, that she was heading towards (or leaving) that shop. Note that a popular shop as defined here has a high flow of shoppers in and out of a shop, rather than the more intuitive notion a popular shop is simply more crowded (a crowded shop with few shoppers entering or leaving the shop is considered unpopular).

In our framework, we will be making the assumption that the people are moving from portal to portal. Based on our observations, the bulk of the shoppers appear to move purposefully rather than saunter around, and we assume that they have specific target destinations in mind. We decided to omit people who are sitting down or loitering in our numerical records because their numbers are negligible compared to those moving around.

Also from our observations, we see that some shops are more popular than others, in that there are more shoppers entering and leaving these shops. For example there are more shoppers

going in and out of a supermarket than a jewelry shop. Besides differences in popularity due to the nature of the shops, we also notice another effect due to the floor areas of the shops, with larger shops being more popular. For example when comparing clothing stores, H&M has a larger area than Zara, which itself is bigger than Mango. Correspondingly, we also see that H&M has highest shopper throughput, while Mango has the smallest.

When traveling between two portals, shoppers have the choice of taking different routes. It would seem logical that certain routes would be preferred to others. The choice of paths is not random but is affected by different factors such as the lengths of paths and how straight they are. We will not assume that we know how these factors influence choices, but aim to learn the relative importance of each factor from our collected data.

We assume the choices of paths taken by different shoppers are independent to keep our modeling framework simple. This models the scenario that shoppers have their own individual interests. The limitation of this independence assumption is that it does not model group behavior, including having the choice of paths being affected by congestion.

Finally, we observe that people are moving at a fairly constant pace and do not frequently slow down or speed up.

4.2.1 Categorization of Vessels

We want to create a model for predicting shopper distribution when the shopping mall is still in a design phase and the occupants of the shop spaces will not yet be known. It is difficult to directly model popularity of each vessel as there are many vessels in a shopping mall and each shop space may be occupied by a huge range of tenants and brands. To simplify the problem, we group the vessels into categories and assume that zoning has been applied to the shopping mall, with each shop space receiving a category label, even if the specific tenant is unknown. We believe that shops in the same category have common factors that influence their

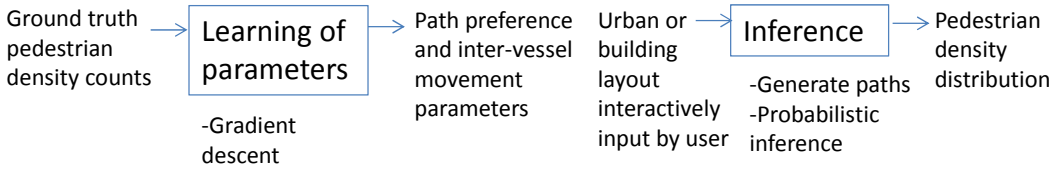


Figure 4.2: Framework for developing and using our pedestrian density distribution model.

popularities beyond floor area. For example the fast food restaurants of Burger King and KFC have similar popularities, which are different from various cosmetic stores, even when all these shops have similar floor areas. It is hoped that these common factors can be learned from data, and having many fewer categories than vessels will make this learning tractable.

Table 4.1 shows the shop categories that we use in our framework. There are more shop categories here than a typical mall directory because the semantic ones used in the mall directory are too broad to have consistent popularity across each category. The details of shop categorization will be discussed in the next chapter, in section 5.4.

4.3 Pedestrian Density Distribution Modeling

4.3.1 Overview

Figure 4.2 shows the process for developing and using our pedestrian density distribution model. There are two parts of the process. First we learn the parameters of our model from ground truth pedestrian counts in the training data. Subsequently we are able to infer crowd density distribution of a new building layout, requiring only the layout design and a shop categorization/zoning plan. The user can then adjust the parameters of the model to interactively find good building layouts.

Categories	**Example stores**
Accessories	Accessorize, Lovisa
Banks	OUB, POSB
Personal care	Body Shop, L'Occitane
Cafés	Spinelli, TCC
Chocolates / wine / flowers	Ferisia De Floral, Sophisca
Cosmetics	Estee Lauder, Chanel
Dental	Q&M, Unity Denticare
DIY stores	Home Fix, Selffix
Electrical	Simplehuman, Harvey Norman
Electronics	Epi Center, I Studio
Fashion	Uniqlo, H&M
Fast food	Burger King, KFC
Foodcourt	Kopitiam, Food Republic
Furniture	Proof, Crate and Barrel
Hairdressers / barbers	EC House, QB House
Health equipment	Osim, Ogawa
Health supplements	GNC, Nature's Farm
Jewelry	Citigems, Gold Heart
Manicure / foot massage	Nail Clinique, The Reflexology Company
Medical clinics	Q&M, Fullerton Healthcare
Money changers	Currency Connect, Dollar Exchange
Pharmacies	Guardian, Unity
Restaurants	Paradise Dynasty, Magosaburo
Shoe/bags	Ecco, Geox
Snacks	Old Chang Kee, Cha Time
Sports	Running Lab, Urban 360
Stationery	kikki.K, Prints
Supermarkets	Cold Storage, Jason
Telcos	Starhub, M1
Travel agents	Global Holidays, JTB
Watches	Swatch, Moments of City Chain

Table 4.1: Shop categories used in this thesis, and corresponding archetypal stores.

There are a number of important variables in our framework. The term ζ is used to represent the id of a node. Each unique potential path taken by a pedestrian is given a path id η, and must be non-looping and non-backtracking. The id of a portal from which a pedestrian starts a trip is θ_{out}, and the trip ends at a portal θ_{in}. The corresponding vessels from which the pedestrian emerges (source) and subsequently enters (sink) are defined by ϑ_{out} and ϑ_{in} respectively. Each vessel belongs to a particular vessel category, where γ_{out} is the category of the source vessel and likewise γ_{in} is the category of the sink vessel.

In our model, various probabilities are related in the following equation:

$$P(\zeta) = \sum_{\substack{\eta, \\ \theta_{in},\theta_{out}, \\ \vartheta_{in},\vartheta_{out}, \\ \gamma_{in},\gamma_{out}}} P(\zeta|\eta)P(\eta|\theta_{in},\theta_{out})P(\theta_{in}|\vartheta_{in})P(\theta_{out}|\vartheta_{out})P(\vartheta_{in},\vartheta_{out}|\gamma_{in},\gamma_{out})P(\gamma_{in},\gamma_{out}) \tag{4.1}$$

where:

- $P(\zeta)$ is the probability of finding a person in a node ζ on a floor layout,
- $P(\zeta|\eta)$ is the probability of finding a person in a node ζ if it is known that the person is taking the path η,
- $P(\eta|\theta_{in},\theta_{out})$ is the probability of a person taking a particular path η given all possible paths between origin portal θ_{out} and destination portal θ_{in},
- $P(\theta_{out}|\vartheta_{out})$ is the probability of a person leaving a vessel ϑ_{out} through portal θ_{out} and likewise $P(\theta_{in}|\vartheta_{in})$ is the probability of a person entering a vessel ϑ_{in} through portal θ_{in},
- $P(\vartheta_{in},\vartheta_{out}|\gamma_{in},\gamma_{out})$ is the probability that the source vessel is $\vartheta_{out}$and the sink vessel is ϑ_{in} if we only knew that these vessels belong to the categories of γ_{out} and γ_{in} respectively, and
- $P(\gamma_{in},\gamma_{out})$ is the probability of finding a person moving from a vessel of category γ_{out} to another vessel of category γ_{in}.

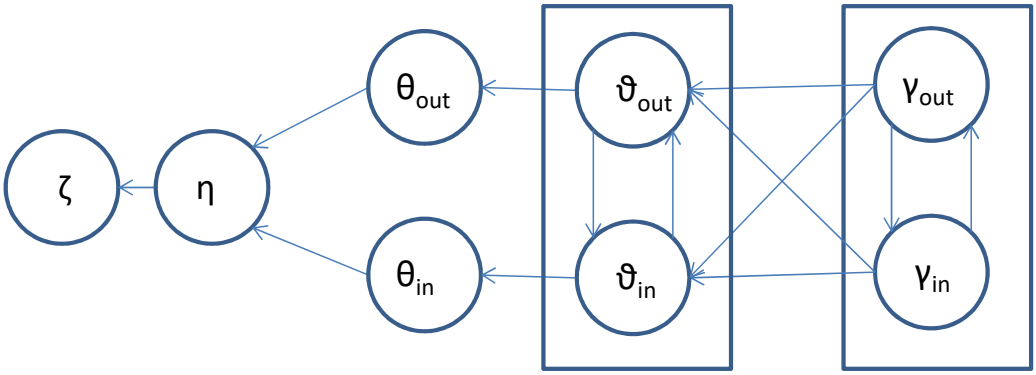


Figure 4.3: Diagram of our model showing the Markov relationships between the variables.

Figure 4.3 shows the Markov relationship between these variables.

If we were to randomly select a pedestrian on a floor of a shopping mall, the term $P(\zeta)$ tells us how likely he will be located on node ζ. The term $P(\eta|\theta_{in},\theta_{out})$ represents the probability of selecting different paths connecting two portals, and encodes the preference of a pedestrian in selecting one path versus another. The term $(P(\theta|\vartheta))$ is the probability of selecting one of the portals connected to a vessel, which intuitively may be thought of as expressing the chance a shopper will choose to use one entrance of a shop versus another, for shops with multiple entrances. The popularities of the vessels are expressed by $(P(\vartheta_{in},\vartheta_{out}))$, which we have to write in joint form because our model assumes that a pedestrian does not start and end at the same vessel in a single trip. However in (4.1), the vessel popularities are expressed conditionally on the vessel categories, because we expect that it is easier to learn the vessel category probabilities $P(\gamma_{in},\gamma_{out})$ than the vessel probabilities directly.

4.3.2 Node Probabilities Conditioned on Paths

Supposed that we know in advance the path that a pedestrian chooses when moving from one portal to another. When the pedestrian is sampled at a particular time instant, where would she be? This depends on the probability $p(\zeta|\eta)$.

In our framework, we make the simple assumption that the pedestrian walks at constant speed from one portal to another. In other words, the probability that she is found in a particular node is

$$P(\zeta|\eta) = \begin{cases} \frac{\text{span}(\zeta)}{\sum_{\zeta_k \in \mathscr{Z}_\eta} \text{span}(\zeta_k)} & , \zeta \text{ lies on the path } \eta \\ 0 & , \zeta \text{ does not lie on the path } \eta \end{cases}$$

where $\text{span}(\zeta)$ returns the longest principal length of the node (defined as the largest absolute eigenvalue when applying Principal Component Analysis (PCA) to the spatial positions of pixels located within the node) and $\mathscr{Z}_\eta$ is the set of all nodes in path η. The principal length of a node in this case is the longest distance between any two points on the edge of the node.

We non-uniformly distribute the node probabilities based on the longest principal length of each node, which is an approximation to the traversal distance in the corresponding region along the path. This caters for a pedestrian taking longer to walk across a larger node, and is more accurate than assuming the nodes are of the same size. In the future, it may be possible to investigate more complex approaches to deal with differing walking speeds, that for example may depend on the crowd densities at various nodes. See figure 4.4.

4.3.3 Path Choice Preference

In cases when there are multiple paths connecting the same pair of portals, we are concerned with the choice of path that a pedestrian may take. The probability that a particular path η is chosen from among those that connect the portals θ_{out} and θ_{in} is denoted $P(\eta|\theta_{in}, \theta_{out})$. See

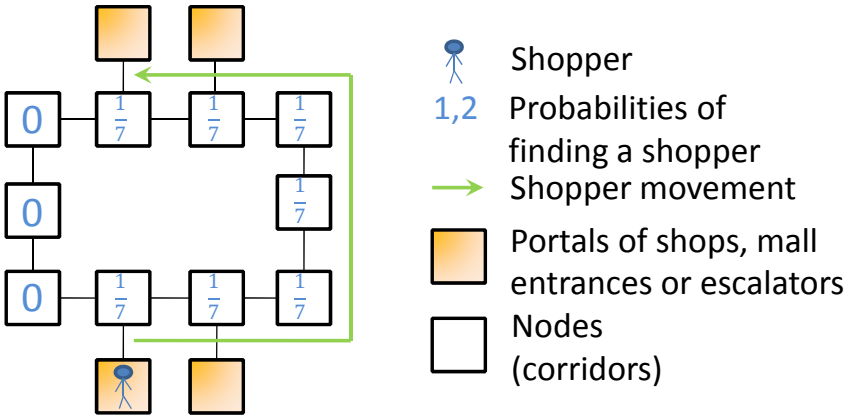


Figure 4.4: Model of a shopper moving from portals to portals. Portals are the shops, escalators and mall entrances. Numbers are the probabilities of finding a shopper in a node given a path ($P(\zeta|\eta)$). For simplicity of illustration, we assume that each node has the same span.

figure 4.5. Note that although in reality there is the possibility that a pedestrian may backtrack and return to a previous node, e.g. when she is lost or changes her mind, for computational feasibility we exclude all paths that include any node more than once.

More interestingly, we are interested in finding out what factors are involved that make a pedestrian prefer one path to another. In our model, we assume that the preferences depend on certain path properties that we call *path descriptors*.

The path descriptors used in our model are:

- **Expanse** of a node is the sum of distances to the nearest walls that can be seen in all directions from the center point of the node, while the expanse of a path is the mean expanse of all nodes along the path. In our implementation, we use 32 possible directions around the center point of a node to compute the expanse.
- **Line of sight** is the mean straight line distance (to the nearest wall) in the direction of walking while traversing the nodes of the path.

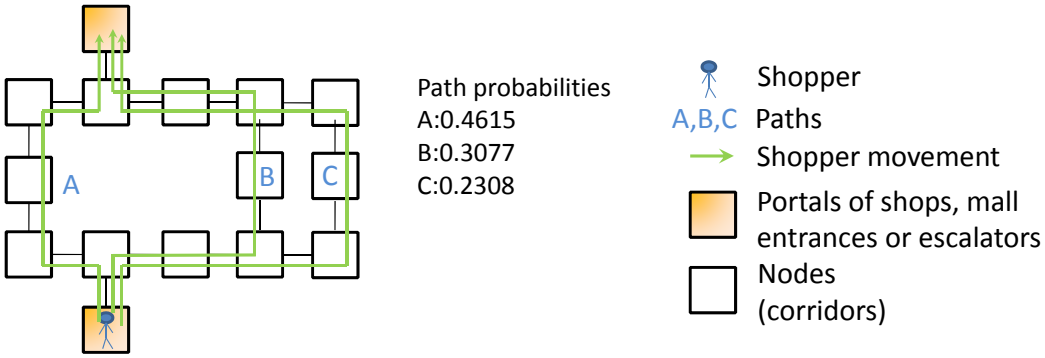


Figure 4.5: Example of three paths to choose from the origin portal to the destination portal. Each path has different probability.

- **Path distance** here is an excess measure, expressed by the length of the path minus the length of the shortest possible path between the source and sink portals.

- **Turn distance** is also an excess measure. If we define total angle of a path as the sum of absolute differential angles of turns executed during movement along the path, the turn distance of a path is its total angle minus the smallest total angle among all alternative paths. Refer to the Figure 4.6.

These descriptors are motivated by and similar to those commonly used to describe free space in buildings in architectural literature [67, 68]. However, instead of using the path descriptors directly, we found that normalization improves performance. Hence for a floor, we consider all acceptable paths and find the mean and standard deviation for each descriptor. Then for each path, we subtract the descriptor values by their corresponding means and divide them by their corresponding standard deviations.

The various path descriptors are used in an expression to derive the conditional probability of

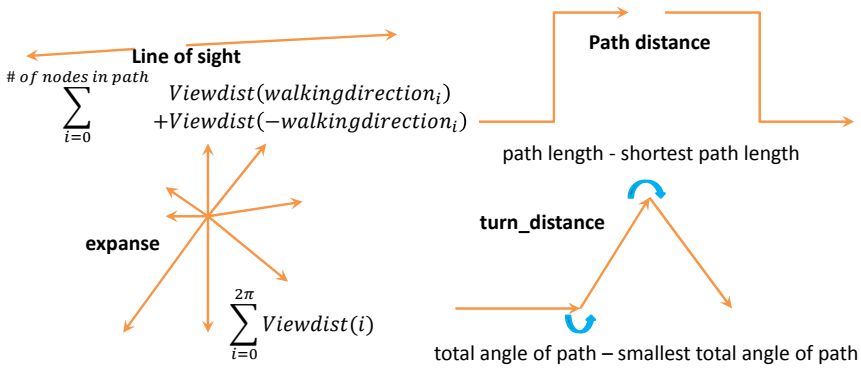


Figure 4.6: Description of how each path descriptor is computed. *Viewdist* function returns the distance to the wall for an input direction. $Walkingdirection_i$ is the direction of walking from current node i to the next node.

choosing a path:

$$P(\eta|\theta_{in},\theta_{out}) = \frac{\Psi(\eta)}{\sum_{\eta_k \in \mathscr{H}} \Psi(\eta_k)}, \tag{4.2}$$

where the function

$$\Psi(\eta) = \sum_{j=1}^{4} a_j \left(1 + \tanh\left(b_j\left(\mathrm{desc}_j(\eta) + c_j\right)\right)\right) \tag{4.3}$$

computes a combined preference of the four path descriptors, namely expanse, line of sight, path distance and turn distance, which are indexed by $i = 1,\ldots,4$. In addition, $\mathscr{H}$ is the set of all paths that start and end with portals θ_{out} and θ_{in}, $\mathrm{desc}_i(\eta)$ computes the value of path descriptor j for path η, while a_j, b_j and c_j are weight parameters (per path descriptor) to be estimated in training. The parameters a_j's, where $a_j \geqslant 0$ and $\sum a_j = 1$, define the relative weighting of different path descriptors. The tanh function operates as a sigmoid to clip large path descriptor values, within which b_j determines the slope at the origin of the sigmoid and c_j the lateral offset.

It is worthwhile comparing this equation to the logistic function more conventionally used for

route choice modeling [124] in transport engineering, given by

$$P_k = \frac{\exp(V_k + B_{CF} \times CF_k)}{\sum_{i \in C} \exp(V_i + B_{CF} \times CF_i)}. \tag{4.4}$$

The term P_k is the probability of choosing route k within a set of paths C, whereas V_k and V_i are utility functions of explanatory variables (equivalent to path descriptors). The terms CF_k and CF_i represent similarity factor between current path and every other path so as to reduce the preference P_k of a path if it is very similar to another path. B_{CF} is a parameter to be estimated.

First we ignore commonality factors (CF terms in the above equation). Second our conditional probability expression comprises a sum of tanh functions with offsets, with each tanh function involving an individual path descriptor. Note that

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{4.5}$$

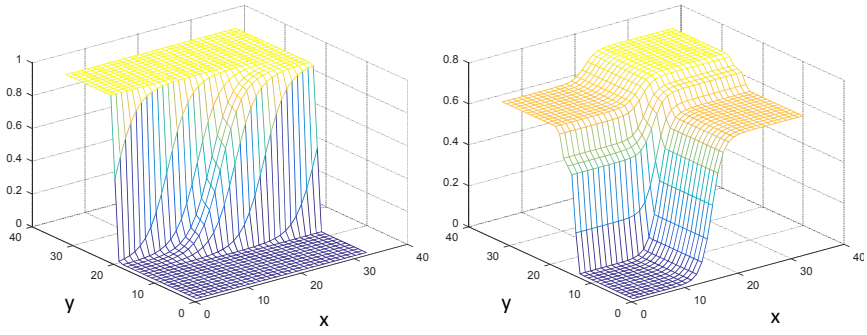
so if we only had a single path descriptor, the conditional probability in (4.2) looks like

$$\frac{1 + \tanh(x)}{1 + \tanh(x) + K} = \frac{2}{K+2} \frac{\exp(2x)}{\exp(2x) + \frac{K}{K+2}} \tag{4.6}$$

where K is the sum of tanh components for all other paths except the current one. This is in a form equivalent to the logistic function in (4.4).

However when using multiple path descriptors our Ψ function becomes a sum of tanh functions. Each path descriptor is passed through a tanh sigmoid before summation.

We will explain why we will prefer to use our route choice function as compared to the original logistic regression route choice function. In the original route choice function using a logistic function, because it combines all path descriptors with only one sigmoid function, the path descriptor with the highest non-linear scale (e.g. x^3 vs x) will automatically dominate. For an example, in the left part of figure 4.7, the change in preference is determined mainly by a single variable y due to its cubic form. In our route choice function, because each path descriptors



$$\frac{\exp(x+y^3)}{1+\exp(x+y^3)} \qquad \frac{1+\tanh(x)+1+tanh(y^3)}{1+1+\tanh(x)+1+tanh(y^3)}$$

Figure 4.7: Comparison of a conventional logistic function and our tanh-based likelihood function using an example in which there are two path descriptors: one that is linear (x-axis), and another that is cubic (y-axis).

are passed through tanh sigmoid before summation, path descriptors with high non-linear scale will be prevented from dominating if the a_j terms are similar. In the right part of figure 4.7, the a_j terms are identical and the variables are prevented from excessively dominating other variables. However, if a path descriptor should ideally dominate if it is truly the only parameter determining path preference, then after training our framework will allow its corresponding a_j term to be close to one, with the other a_j terms zero. Hence this route choice function form provides greater flexibility.

We do not apriori define the a_j, b_j and c_j parameters but instead learn them from ground truth pedestrian counts as described in section 4.5. In particular, we differ from existing approaches in which route choice parameters are learned from GPS data. Depending on the parameters, the preferences for various paths may increase or decrease as path descriptor values change as shown in the figure 4.8. It is designed in such a way because most path descriptors do not increase or decrease linearly with path preference. For example suppose a pedestrian wants to go to a destination and has a choice of two routes, one which is 5 meters long and another 15 meters. Intuitively we expect the pedestrian to strongly prefer the shorter route. If the pedes-

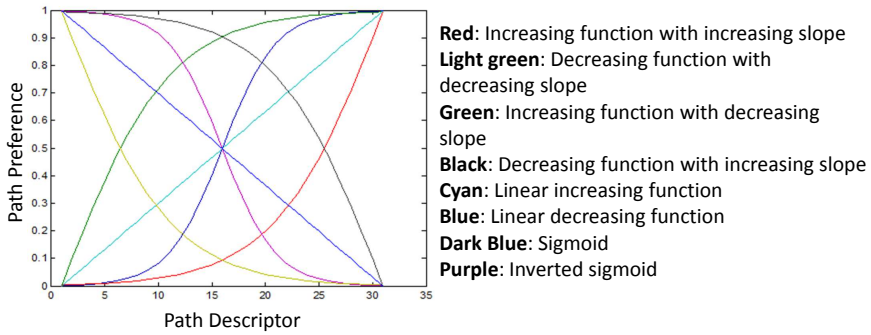


Figure 4.8: The tanh function allows modeling of different rates of change of path preference with respect to path descriptor values.

trian wants to go to a further destination and also has a choice of two routes, but now one is 100 meters long and the other 110 meters. Although the difference in lengths are identical to the first scenario, namely 10 meters, we expect that the pedestrian will have a weaker preference for the shorter route. This is because the pedestrian is less likely to perceive the relative difference of effort in walking the two routes and may choose the longer route because perhaps it has a wider walking path and fewer turns. In our model, this intuition may manifest as a slow down in the decrease of path preference as path distance increases (e.g. yellow curve in figure 4.8).

4.3.3.1 Regularization with Path Preference Priors

One concern of learning path preference parameters directly from data is that the amount of data we have may be very limited. Hence we also explore the use of prior regularization to mediate this problem. The hope is that the learning algorithm will not overfit the data if the

path preference parameters are interpolated with priors. The equations

$$\begin{aligned} b'_j &= \tan\left(\beta \tan^{-1}(b_j) + (1-\beta)\tan^{-1}(1)\right) && \text{, for } j = 1,2 \\ b'_j &= \tan\left(\beta \tan^{-1}(b_j) + (1-\beta)\tan^{-1}(-1)\right) && \text{, for } j = 3,4 \\ c'_j &= \beta c_j + (1-\beta)\times 0 \\ a'_j &= \beta a_j + (1-\beta)\times 1 \end{aligned} \tag{4.7}$$

show how we interpolate path preference parameters to regularizing priors, where β is the interpolation parameter. The parameter β ranges from 0 to 1, where $\beta = 0$ means only the priors are used and $\beta = 1$ means only the learned path preference parameters are used. The priors are chosen to reflect the expectation that expanse and line of sight will be positively correlated with path preference, while path and turn distances will be negatively correlated, based on norms from architectural theory [67, 68]. The best way to interpolate the slope parameters **b** is to linearly interpolate the angles of the slopes, rather than the **b** values themselves; this explains the use of the tangent functions in (4.7).

4.3.4 Conditional Portal Popularity

To cater for shops with more than one entrance, our model includes the terms $P(\theta_{in}|\vartheta_{in})$ and $P(\theta_{out}|\vartheta_{out})$ which specify the probability of selecting one of the portals when a pedestrian enters or leaves a particular vessel.

The probability of selecting one of these portals θ given a vessel ϑ is given by:

$$\begin{aligned} &P(\theta|\vartheta) \\ &= \begin{cases} \frac{1}{|\Theta_\vartheta|} & \text{, } \theta \in \Theta_\vartheta \\ 0 & \text{, otherwise} \end{cases} \end{aligned} \tag{4.8}$$

where Θ_ϑ is the set of all portals belonging to vessel ϑ. Basically, we define equal chance of selecting any portal associated with the vessel. This is a simplification as e.g. the main entrance

to a shop will more likely be used than a side entrance. In the future, we may cater for more complex vessel layouts where each portal may be assigned a different level of preference.

A further simplification adopted is the assumption that all portals are bidirectional, i.e. they can serve as either origin or destination portals. This is generally true except for portals associated with escalators, which are unidirectional. However, we chose to ignore this complexity at this time because many, but not all, escalators come in adjacent pairs with complementary directions. In the future, we can enhance our model to address this limitation.

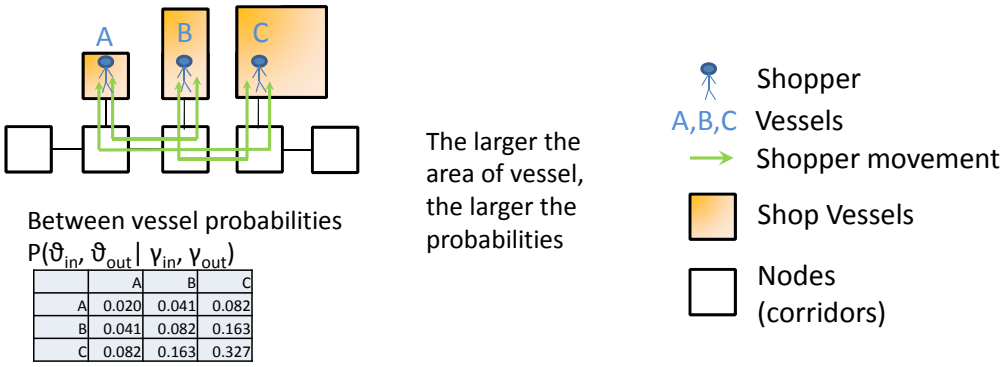
4.3.5 Conditional Vessel Popularity

Within our pedestrian density model, there is the term $P(\vartheta_{in}, \vartheta_{out} | \gamma_{in}, \gamma_{out})$ which represents the probability of randomly selecting a pedestrian who is moving between the vessels ϑ_{out} and ϑ_{in}, assuming we knew in advance the vessel categories to which the vessels belong.

In order to model this probability, we treat vessel categories that are associated with internal floor space (generally shops) differently from vessels that are essentially just gateways (escalators, mall entrances). We call the former *area vessels* and the latter *non-area vessels*.

For area vessels, we assume that the popularity of a vessel is proportional to its corresponding floor area. For example a cafe that is twice as big as another will be twice as popular. This is of course not completely accurate because a cafe may be more popular because it serves better food. However when trying to predict popularity prior to actually knowing the specific shop that will lease a space, this seems to be a reasonable heuristic. A more nuanced model of the popularities of area vessel categories taking into account the shop categories of the area vessels will be explored in the next chapter.

For non-area vessels, we assume that each vessel within the same category is equally popular. In order to cater for gateways that are designed to have different throughput, e.g. main ver-



	A	B	C
A	0.020	0.041	0.082
B	0.041	0.082	0.163
C	0.082	0.163	0.327

Figure 4.9: Example of movements between vessels. $P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out})$ represents probability of finding a pedestrian moving between vessels from vessel categories $(\gamma_{in}, \gamma_{out})$.

sus side entrances of malls, we have more refined categories; for example, high-volume mall entrances is a different category to low-volume mall entrances.

In addition, we enforce the requirement that ϑ_{in} cannot be the same as ϑ_{out}. In other words, a pedestrian will not leave a vessel, wander circularly and return to it. Once again this may not be perfectly true in reality but is expected to a reasonable simplification.

Figure 4.9 shows probabilities of choosing each pair of source and sink vessels out of all possible pairs that fall within the appropriate vessel categories. The probabilities depend on the areas of the corresponding shops.

The probability of moving from vessel ϑ_{out} to vessel ϑ_{in}, conditioned on knowing the vessel categories, may be expressed as

$$\begin{aligned} P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out}) &= P(\vartheta_{in}|\vartheta_{out}, \gamma_{in}, \gamma_{out})P(\vartheta_{out}|\gamma_{in}, \gamma_{out}) \\ &= P(\vartheta_{in}|\vartheta_{out}, \gamma_{in})P(\vartheta_{out}|\gamma_{out}) \end{aligned} \tag{4.9}$$

by applying the appropriate conditional independencies, with the component terms given by

$$P(\vartheta_{in}|\vartheta_{out},\gamma_{in}) = \begin{cases} \frac{\text{area}(\vartheta_{in})}{\left(\sum_{\vartheta_k \in \mathscr{V}_{in}} \text{area}(\vartheta_k)\right) - \text{area}(\vartheta_{out})} & ,\text{cat}(\vartheta_{out}) = \gamma_{in} \wedge \vartheta_{in} \neq \vartheta_{out} \wedge \text{cat}(\vartheta_{in}) = \gamma_{in} \wedge \{\vartheta_{out}\} \neq \mathscr{V}_{in} \\ \frac{\text{area}(\vartheta_{in})}{\sum_{\vartheta_k \in \mathscr{V}_{in}} \text{area}(\vartheta_k)} & ,\text{cat}(\vartheta_{out}) \neq \gamma_{in} \wedge \text{cat}(\vartheta_{in}) = \gamma_{in} \\ 0 & ,\text{otherwise} \end{cases} \tag{4.10}$$

and

$$P(\vartheta_{out}|\gamma_{out}) = \begin{cases} \frac{\text{area}(\vartheta_{out})}{\sum_{\vartheta_k \in \mathscr{V}_{out}} \text{area}(\vartheta_k)} & ,\text{cat}(\vartheta_{out}) = \gamma_{out} \\ 0 & ,\text{otherwise.} \end{cases} \tag{4.11}$$

Here $\text{area}(\vartheta)$ returns the area of the vessel ϑ (or unity for a non-area vessel), $\text{cat}(\vartheta)$ returns the category to which the vessel ϑ belongs, $\mathscr{V}_{in}$ is the set of vessels that belong to vessel category γ_{in}, and likewise $\mathscr{V}_{out}$ is the set of vessels that belong to vessel category γ_{out}.

Generally we define the popularity of an area vessel to be proportional to its area, but the equations are more complex for movement between vessels of the same category because the model prevents a pedestrian from leaving and returning to the same vessel (this is not a problem for movement between different categories). In this case, we exclude the area of the source vessel in computing the popularities of other vessels with the same category, except when the category contains only one vessel in which case the probability is set to zero. For non-area vessels, the same equations apply except the $\text{area}(\vartheta)$ function evaluates to unity. This way, all vessels in a non-area vessel category are assumed to be equiprobable.

4.3.6 Vessel Category Popularity

As stated previously, the purpose of having vessel categories is to try and develop a more generalized model of popularity that is not tied to specific vessels, e.g. particular brands of

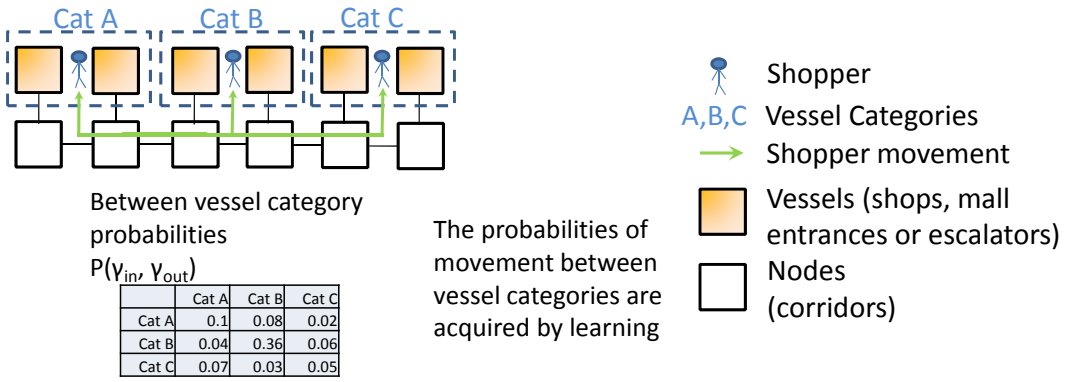


Figure 4.10: Example of probability of finding a person moving from one vessel category to another vessel category.

shops. This would allow us to learn a model that can apply more widely to predict pedestrian density distributions in new mall layouts with category zoning, without having to know the actual shops first.

In this chapter, we attempt to learn the popularity parameters of non-area vessel categories, while assuming that the relative popularities of area vessel categories are still based on shop areas. A more nuanced model of the popularities of area vessel categories will be explored in the next chapter.

Figure 4.10 shows that probabilities of finding a person moving from vessel category γ_{out} to vessel category γ_{in} out of all possible vessel category pairs.

The probability of moving from a vessel category γ_{out} to another vessel category γ_{in} is the term $P(\gamma_{in}, \gamma_{out})$, which is defined in our model as

$$P(\gamma_{in}, \gamma_{out}) = \begin{cases} 0 & , \gamma_{in} = \gamma_{out} \wedge \gamma_{in} \in \mathscr{S} \wedge \gamma_{out} \in \mathscr{S} \\ P(\gamma_{in})P(\gamma_{out}) + K_1(\gamma_{in}, \gamma_{out}) + K_2(\gamma_{in}, \gamma_{out}) & , \text{otherwise} \end{cases} \tag{4.12}$$

where $\mathscr{S}$ is the set of all vessel categories that each contains only one vessel. The functions

K_1 and K_2 are defined as

$$K_1(\gamma_{in}, \gamma_{out}) = \begin{cases} \frac{1}{2}P(\gamma_{in})^2 \frac{P(\gamma_{out})P(\gamma_{in})}{\sum_{\gamma_k \neq \gamma_{in}} P(\gamma_k)P(\gamma_{in})} & , \gamma_{in} \in \mathscr{S} \\ 0 & , \text{otherwise} \end{cases}$$
$$K_2(\gamma_{in}, \gamma_{out}) = \begin{cases} \frac{1}{2}P(\gamma_{out})^2 \frac{P(\gamma_{in})P(\gamma_{out})}{\sum_{\gamma_k \neq \gamma_{out}} P(\gamma_k)P(\gamma_{out})} & , \gamma_{out} \in \mathscr{S} \\ 0 & , \text{otherwise.} \end{cases} \tag{4.13}$$

In our model, if γ_{in} and γ_{out} are not single-vessel categories, then we assume that they are independent, i.e. $P(\gamma_{in}, \gamma_{out}) = P(\gamma_{in})P(\gamma_{out})$. However if say γ_{in} is a single-vessel category, we know that γ_{out} cannot be the same category, otherwise it will lead to a situation in which the source and sink vessels are identical. In this case, we set $P(\gamma_{in} = \gamma_{out})$ to zero rather than $P(\gamma_{in})^2$ (as per the independence assumption), and proportionally distribute that probability to other pairings of source categories with γ_{in}. We do so likewise when γ_{out} is a single-vessel category.

In this chapter, the probability for an individual category γ is computed as

$$P(\gamma) = \begin{cases} \frac{\sum_{\vartheta_k \in \mathscr{V}_\gamma} \text{area}(\vartheta_k)}{\sum_{\vartheta_j \in \mathscr{V}_a} \text{area}(\vartheta_j)} \left(1 - \sum_{\gamma_k \in \mathscr{C}_n} f_{\gamma_k}\right) & , \gamma \text{ is an area vessel category} \\ f_\gamma & , \text{otherwise} \end{cases} \tag{4.14}$$

subject to

$$\sum_{\gamma_k \in \mathscr{C}_n} f_{\gamma_k} \leq 1$$
$$f_\gamma \geq 0 \tag{4.15}$$

where $\mathscr{V}_\gamma$ is the set of vessels of category γ, $\mathscr{V}_a$ is the set of all area vessels (on the particular floor) and $\mathscr{C}_n$ is the set of all non-area vessel categories. The term f_γ is the popularity of the non-area vessel category γ, that will be learned in our framework as described later. There are four categories of non-area vessels, namely low volume escalators, high volume escalators, low volume mall entrances and high volume mall entrances, each with its own f_γ value. For area vessel categories, the popularities are partly proportional to the total areas of their respective vessels, and partly taking into account the learned popularities of non-area vessel categories.

4.4 Likelihood of Pedestrian Counts

The probabilities in the previous sections are not directly observable. What are observable are physical pedestrian counts, which may be considered as samples drawn from some $P(\zeta)$ distribution of pedestrian densities across the nodes.

Conversely, if we have actual pedestrian counts and an estimated $P(\zeta)$ distribution, we may pose the question: what is the likelihood that the observed counts are drawn from the estimated distribution?

Suppose there are N nodes on the floor with each node having a label $\zeta_i, i = 1,\ldots,N$. Given a fixed number T of pedestrians on a floor, the distribution of these pedestrians to different nodes follows a multinomial distribution:

$$P(\mathbf{m}) = P(m_1,\ldots,m_N) = \frac{T!}{m_1!\ldots m_N!} p_1^{m_1}\ldots p_N^{m_N} \tag{4.16}$$

where p_i is $P(\zeta_i)$ representing the probability of finding a pedestrian in node ζ_i and m_i is the number of pedestrians in node ζ_i. We require $\sum_i m_i = T$ because the sum of pedestrian counts across all nodes must be the total number of pedestrians.

In subsequent sections, the multinomial distribution $P(\mathbf{m})$ will be used for learning model parameters and for evaluating the goodness of fit to ground truth observations.

4.5 Learning

In this section, we want to find the best parameters of our model such that the estimated pedestrian density distribution best fits the observed ground truth pedestrian count for each node. For a floor layout with N nodes, the pedestrian density distribution is represented by $P(\zeta_i), i = 1,\ldots,N$, where each $P(\zeta_i)$ is modeled by (4.1). The parameters that we want to

optimize are path preference parameters **a**, **b** and **c** which are vectors respectively containing a_j, b_j and c_j with $j = 1,\ldots,4$ as expressed in (4.3), as well as the non-area vessel category popularities **f** which is vector containing $f_\gamma, \forall \gamma \in \mathscr{C}_n$ as per (4.14).

To measure the goodness of fit of the pedestrian density distribution to the ground truth node pedestrian counts given by $\mathbf{m}'$ with a total ground truth pedestrian count T', we can compute the likelihood of the overall multinomial distribution generating the ground truth counts:

$$P(\mathbf{m} = \mathbf{m}'; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{f}) = T'! \prod_{i=1}^{N} \frac{p_i^{m_i'}}{m_i'!} \tag{4.17}$$

where the p_i's depend on **a**, **b**, **c** and **f**.

Since the multiplicative constants do not affect the optimization of the parameters, we can instead define a simplified log likelihood function:

$$L(\mathbf{m}'; \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{f}) = \sum_{i=1}^{N} m_i' \log(p_i). \tag{4.18}$$

When dealing with multiple floors, a challenge is that the total ground truth pedestrian counts T' are different for different floors. Since we are determining a single set of model parameters that can apply across different floors, we need a way to combine the likelihoods of different floors. A direct summation of the log likelihoods will however unduly weigh the model parameters to better fit the floors that had larger pedestrian counts. Since we are creating a model that can be used for predicting pedestrian density distributions in new layouts without knowing the total pedestrian counts, the approach taken is to normalize the ground truth counts such that all floors have the same total number of pedestrians. The (negated) joint log likelihood function that is used as the cost function is

$$\lambda = -\sum_{\text{floor}} \frac{L^{\text{floor}}}{T'^{(\text{floor})}}. \tag{4.19}$$

To learn the parameters, we will minimize λ with respect to the parameters **a**, **b**, **c** and **f** using the gradient-based BFGS method. The constraints used in optimization are $f_\gamma \geq 0, \forall \gamma \in \mathscr{C}_n$ which ensure that the probabilities learned for non-area vessel categories are non-negative.

Prior to optimization, we have to enumerate all non-looping paths between all pairs of origin and destination portals for each floor. This is done through a basic depth-first search algorithm. We then compute the path descriptor values for each path.

Interleaved optimization of the path preference parameters $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ and non-area vessel category popularities **f** is carried out for computational efficiency. We first minimize λ with respect to $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ of our model, keeping **f** fixed. In the next step, we find the optimal **f** while keeping $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ fixed. Convergence is typically reached in 5 cycles.

Similar to section 4.3.3.1, the gradient descent algorithm initialization of path preference parameters is based on norms from architectural theory [67, 68]. The $b_j, j = 1, \ldots, 4$ terms are initialized as 1, 1, -1 and -1 for expanse, line of sight, path distance and turn distance respectively, because expanse and line of sight are expected to correlate positively with path preference, while path distance and turn distance are expected to correlate negatively. The terms c_j are set to zeroes and a_j set to ones.

Detail steps of the algorithms are shown in the appendix A.2.

4.6 Inference

Once we have learned the parameters of our model, we may want to estimate the pedestrian density distribution for a new floor layout design. Given a new floor layout and vessel category labels as input, we generate all possible non-looping paths and compute the path descriptors. Then using the path preference parameters and the popularities of non-area vessel categories that were learned, we can compute $P(\zeta)$.

The main computational cost of our method is in generating all the paths through depth-first search. If there is a subsequent change in the geometry of the layout but not in the topology, we only need to recalculate the path descriptor values which is fast. The cost of changing vessel category labels is also small. However if the topology is changed, we have to re-generate the paths.

4.7 Evaluation Methodology

4.7.1 Data Collection of Pedestrian Counts

Pedestrian count data may be ideally collected with the aid of surveillance videos. However, access to such videos in public shopping malls is hard to obtain. Instead we collected pedestrian counts by taking our own videos using an Apple iPad Air. As the videos were recording, we walked through all floors of two Singapore shopping malls (Ion [2] and Novena Square [4]), twice for each floor. This method is based on the instruction manual by Space Syntax limited for collection of pedestrian counts [148]. Counting of pedestrians in the video is conducted using the *static snapshots* method rather than *gate counting* method as it is faster and can be collected by a single person. We also obtained the layouts of the shopping malls from their websites.

We collected data from eight floors of Ion and three floors of Novena Square on Saturday 23 August 2014. For the Ion shopping mall, we captured videos starting at 4:10pm and ending at 5:40pm, while for the Novena Square shopping mall the video collection was from 5:55pm to 6:30pm. We obtained a pedestrian count per node based on the number of people appearing in the corresponding region in the videos. The two passes provide two pedestrian counts per node that we reduced to an average count per node. The mean numbers of people on each floor in our dataset are 37.5, 42.5, 46.5, 103, 145, 182.5, 286 and 520.5 for Ion shopping mall, while for the Novena Square shopping mall the numbers are 17, 67.5 and 171.5.

Note that these counts were obtained manually. Although we would like to use our pedestrian detector described in chapter 3, which would have been fine in many frames of our collected data, there will also a number of places in which the crowd density is high with substantial occlusion of various pedestrians that would have been missed by our detector. These frames would require manual labeling, and for consistency and accuracy we decided to manually count the pedestrians for all nodes.

An assumption made in interpreting the data is that the total number of people on the floor is the sum of the average pedestrian count for each node. This assumes that no pedestrian has been missed in the data collection, and that there is also no double counting. However this may not be entirely true due to the sequential manner in which the video camera is moved.

The dataset contains 11 floors of shopping mall data from two different shopping malls. Each floor has about 20 to 49 nodes. We collected two counts of pedestrian density counts for each node. We also collected the number of people entering and exiting each shop for two of the floors.

4.7.2 Baseline Prediction

Although we have a model that predicts the pedestrian density distribution $P(\zeta_i), i = 1,\ldots,N$, but we are unable to compare this to the ground truth distribution which is hidden. What we can observe are pedestrian counts, which may be considered as samples from this hidden distribution. Hence to evaluate the accuracy of our model, we have to do a relative comparison to a baseline distribution.

We use a baseline distribution in which all the probabilities $P(\zeta_i)$ are identical, i.e. a uniform multinomial distribution. In the absence of prior knowledge, this baseline is mathematically the best guess that can be made, in the sense that it has the highest expected likelihood value

from counts sampled from any possible multinomial distribution (with the same number of nodes). Further details are provided in appendix A.1.

4.7.3 Log Likelihood Ratio

To compare our modeled distribution to the baseline for a single floor, we use the log likelihood ratio. For multinomial distributions, we may express this as:

$$R = \log\left(\frac{\frac{T'!}{\prod_{i=1}^{N} m_i'!}\prod_{i=1}^{N} q_i^{m_i'}}{\frac{T'!}{\prod_{i=1}^{N} m_i'!}\prod_{i=1}^{N} p_i^{m_i'}}\right) = \sum_{i=1}^{N} m_i' \log(q_i) - m_i' \log(p_i) \tag{4.20}$$

where for node i: p_i is $P(\zeta_i)$ which is obtained from the learned **a**, **b**, **c** and **f** parameters, $q_i = 1/N$ is the baseline probability parameter and m_i' is ground truth count. In addition, T' is total number of people observed on the floor, while N is number of nodes for the floor.

Note that in this form, a zero value for R indicates that both distributions are equally likely, while the more negative the value the more likely is our modeled distribution.

In the evaluation, we do not normalize the ground truth pedestrian counts, as we did in the training phase, since this is a more accurate representation of the actual performance.

4.7.4 Neyman-Pearson Test

Although the log likelihood ratio is a comparison of two distributions in generating the ground truth pedestrian counts, it should be noted that the pedestrian count is itself a random sample from a hidden ground truth distribution.

If we obtain a favorable log likelihood ratio, it is useful to find out if this occurrence is indicative that the model distribution is a significantly better approximation to the ground truth

distribution than the baseline, or if such an occurrence happens with reasonable chance even when the baseline distribution is the better approximation.

One way to analyze this is to conduct hypothesis testing that exploits the Neyman-Pearson Lemma [104]:

- The first stage involves generating multiple samples taken from a Dirichlet-multinomial distribution with alpha parameters all set to one, which corresponds to sampling from some random member from the family of multinomial distributions. In our case, given the known number of nodes in a floor layout, together with the intended total number of pedestrians T, we obtain a random sample of pedestrian counts at each node. By running through this sampling pipeline multiple times, we obtain synthetic pedestrian counts that come from many different possible multinomial distributions.
- The second stage uses the randomly synthesized pedestrian counts from the first stage, and computes the multiple log likelihood ratios comparing the baseline and our model hypothesis. The proportion of synthetic log likelihood ratios that outperforms the measured log likelihood ratio (using the ground truth pedestrian counts) is noted as the *p-value* — if this p-value is very low, it means that our model hypothesis performs better than the baseline *only* for the ground truth pedestrian counts and not the synthetic counts, which indicates that it is significantly sensitive to the ground truth distribution. Conversely, a high p-value means that the model hypothesis is not significantly different from the baseline, and the favorable log likelihood ratio was a chance occurrence.

The algorithm is presented more precisely below.

Neyman-Pearson Test

Input:

N: Number of nodes in the floor layout

$p_1 \dots p_N$: The $P(\zeta_i)$ values computed from our model parameters

T': Total number of pedestrians on the floor

$\mathbf{m}'$: Ground truth pedestrian counts per node

Procedure:

Initialize the counters $n_{pass} = 0$ and $n_{fail} = 0$.

Find the log likelihood ratio of generating ground truth counts $\mathbf{m}'$ from baseline and model distributions. Set this value as the threshold

$$t = \sum_{i=1}^{N} m_i' \log(\frac{1}{N}) - m_i' \log(p_i).$$

Set $\alpha = [1, \dots, 1]$.

for 100,000 iterations **do**

Draw a synthetic count sample $\mathbf{s}$ from a Dirichlet-multinomial distribution $\mathrm{DM}(T', \alpha)$.

Find the log likelihood ratio of generating $\mathbf{s}$ from baseline and model distributions

$$u = \sum_{i=1}^{N} s_i \log(\frac{1}{N}) - s_i \log(p_i).$$

if $u < t$ **then**

Increment n_{pass} by 1.

else

Increment n_{fail} by 1.

end if

end for

Return p-value $= \frac{n_{pass}}{n_{pass}+n_{fail}}$.

The sampling procedure is repeated 100,000 times. For a graphical interpretation of the test, refer to appendix A.4.

4.7.5 Fisher's Method for Meta Analysis

In our experiments, we independently predicted pedestrian density distributions for a number of different floor layouts, from which we conducted Neyman-Pearson testing to obtain a p-value per floor. Although this tells us the significance of the model prediction for each floor, we want to know if the model can make systematically significant predictions across many different floor layouts.

One approach to investigate this is to use Fisher's method [49]. The purpose of using this method is to determine the probability that the null hypothesis is simultaneously true across all our independent tests. The underlying assumption of Fisher's method is that if the null hypothesis is true for F number of tests, the p-values, $\rho_k, k = 1, \ldots, F$ are uniformly distributed between 0 and 1. Hence the negative logarithm of the p-values, $-\ln(\rho_k)$, follow an exponential distribution, while $-2\ln(\rho_k)$ will follow a chi-squared distribution with 2 degrees of freedom. The sum of F such independent random variables will follow a chi-squared distribution with $2F$ degrees of freedom, so

$$Z = -2\sum_{k=1}^{F} \ln(\rho_k) \sim \chi^2_{2F} \tag{4.21}$$

The meta null hypothesis is therefore that Z is drawn from this chi-squared distribution. Using our actual measured p-values for different floors, say we obtain Z'. The meta p-value ρ_{meta} is given by

$$\rho_{\text{meta}} = P(Z \geq Z') \tag{4.22}$$

which is the probability that the meta null hypothesis is true when Z' or higher is observed. This probability may be computed from statistical software.

4.7.6 Experimental Setup

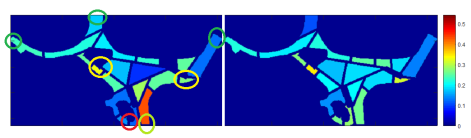
To evaluate prediction performance of our model, we choose one floor as the test set and the rest as the training set. We rotate through each floor in turn as the test set and others as the training set, leading to leave-one-out cross validation (LOOCV). We used this method of partitioning the data into training and test set as we only have eleven floors of data. We report log likelihood ratios comparing a baseline uniform distribution to $P(\zeta)$ estimated by our model. We also evaluate our model using the Neyman-Pearson test and Fisher's method. The machine we used in all experiments was running Windows 7 on a Intel 2 GHz i7 processor and had 4 GB RAM.

4.8 Experimental Results

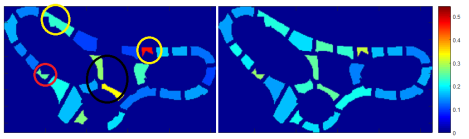
In this section, we will show visual and quantitative pedestrian density prediction results. For clarity of presentation, we first show the visual and quantitative results based on the optimal prior mixing of path preference parameters, in which $\beta = 0.2$. Subsequently we will present results showing how this optimal β value is obtained.

4.8.1 Leave-One-Out Cross Validation Results

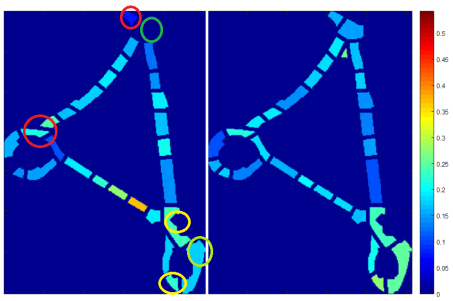
To visually compare ground truth pedestrian counts with a predicted pedestrian density distribution, we first process the ground truth counts by dividing the number of pedestrians found in each node by the total number of pedestrians on the floor, which may be considered as converting the ground truth counts into a form of ground truth density. Both the ground truth and predicted densities for each node are further normalized by the floor area of the node, to provide an intuitive sense of the human density per unit area. For improving the perceptual quality of visualization, these values are further transformed monotonically prior to rendering



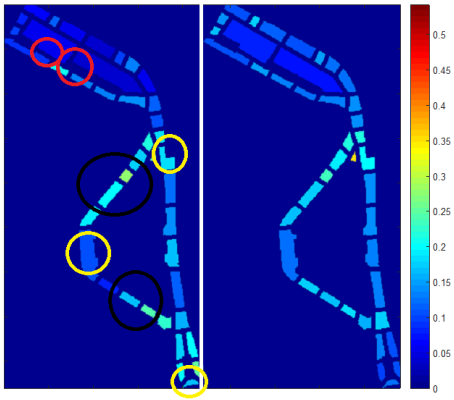
(a) Ion Floor 1 (Pearson correlation coefficient=0.6980)



(b) Ion Floor 4 (Pearson correlation coefficient=0.4714)



(c) Ion Floor B1 (Pearson correlation coefficient=0.3690)



(d) Ion Floor B4 (Pearson correlation coefficient=0.4472)

Figure 4.11: Visual comparison of ground truth and predicted density distributions. Left of each pair: ground truth densities. Right of each pair: predicted densities. Further annotations: red circles are low volume escalators, yellow circles are high volume escalators, dark green circles are low volume mall entrances, light green circles are high volume mall entrances, and black circles are common corridors or intersections. Note that the numbers in the color legends do not directly map to density values, as they have been monotonically transformed to improve perceptual quality of visualization. (Has to be viewed in color.)

as colors in the floor maps, but are not otherwise used in our quantitative results. Figure 4.11 shows visualization of the ground truth and predicted pedestrian densities for a number of floor layouts. Each floor layout is visualized as a pair: the ground truth densities are shown on the left of each pair, while the predicted densities are shown on the right of each pair. The complete set of visual results can be found in appendix A.3.

We computed the Pearson correlation coefficients between the model and ground truth densities, and the results were 0.6980, 0.4714, 0.3690 and 0.4472 for floors 1, 4, 5 and 8 of the Ion shopping mall respectively. From the coefficients, there is moderate correlation between the model and ground truth densities for all the four layouts shown. While the model is able to

accurately predict most parts of the layouts, in particular if we analyze locations where we expect to have difficulty modeling due to significant density variations across different instances, namely mall entrances and escalators, we see that our model is still able to reasonably predict the densities at these locations — these are useful for assessing the accuracy of our model. For example for the floors in figure 4.11(a), (b) and (c), our model accurately predicts the densities at most locations, including around most of the escalators and mall entrances. One exception is the mall entrance at the bottom of the layout in figure 4.11(a) in which our model predicts high traffic, but not as high as ground truth. This is probably because the entrance has exceptionally high traffic as it leads to a train station, a situation which occurs rarely in the training dataset.

Our model appears able to accurately predict higher densities around intersections, such as the Y-junctions located near the centers of figure 4.11(b) and (d). Note that this is an interesting emergent effect as our model does not explicitly enforce intersections to have higher densities.

Conversely, our model generally predicts remote regions to have lower densities which matches well with the ground truth, as typified by the top-left and right-most regions in figure 4.11(b). This is also true for the remote top regions of figure 4.11(c) and (d), for which the lower density predictions are relatively accurate despite the presence of low volume escalators. However, the remote bottom region of figure 4.11(d) is under predicted by our model despite the presence of a high volume escalator. This may be due to an underestimation of shop popularities in that region.

Our model is also able to accurately predict high traffic in connecting corridors between high volume portals. Examples of such patterns include the regions marked in black circles in figure 4.11(d).

4.8.2 In-Test Optimization

One of the observations made during our experiments is that the learned popularities of the non-area vessel categories $f_\gamma, \forall \gamma \in \mathscr{C}_n$ vary substantially from training batch to batch, and even more so from floor to floor. This implies that there is perhaps no consistent popularity that can be predicted for non-area vessel categories. For example the high volume escalators on a floor may have substantially different traffic to the high volume escalators on a different floor.

Our conjecture is that the popularities of mall entrance and escalator vessels are affected by a huge set of factors and are hard to generalize across malls and floors. There is a significant dependence on where the portals of these vessels connect to, with considerations including proximity to residential buildings, public transport, carparks and other shopping malls. They also vary with the different layouts of the connecting floors. Since these parameters are difficult to learn and predict, we alternatively envisage an application scenario in which the architect, being much more aware of various external factors, may interactively set the popularities of different escalators and mall entrances, and visually interpret how these affect the pedestrian density distributions in near real time.

For figure 4.12, we see that the circled regions are under predicted and it can be improved if the predicted popularities of the escalators in these regions were ramped up. Hence we additionally conducted such in-test optimization of the popularities of non-area vessel categories to find out what are the best test results obtainable if the popularities of non-area vessels were predicted perfectly.

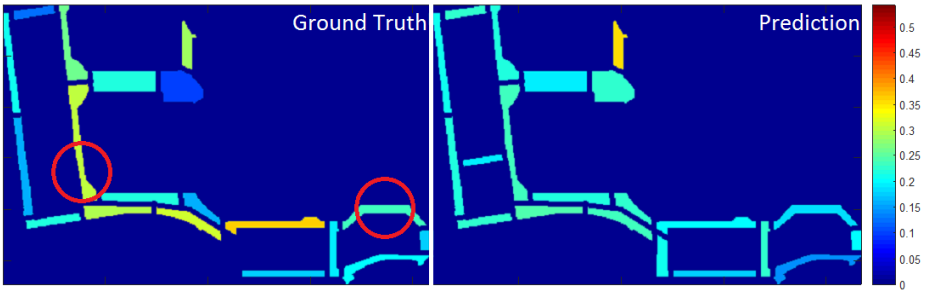


Figure 4.12: Example where pedestrian density prediction can be improved by ramping up the popularities of escalators in circled regions.

In order to visualize the effect of varying the popularities of non-area vessel categories (mall entrances and escalators), we gradually increment the popularity f_γ of the particular non-area category from 0 to 1, and then compensate for the popularities of the other categories to maintain having $P(\gamma_{in}, \gamma_{out})$ sum to 1. Figure 4.13 shows the gradual change of the predicted pedestrian density distribution as f_γ is incremented (left to right). From this visualization, we can see how the probabilities of different floor nodes are affected by the popularities and placements of different non-area vessels, and also depend on the floor layout. When we increase the popularities of the low volume escalators (red circled region), pedestrian densities are predicted to increase near the escalator-connected node, while reducing in further nodes. Notice that when the popularity of the low volume escalator is increased, the Pearson correlation coefficient drops, which is logical which this is effectively converting a low volume escalator into a high volume one. Conversely when we increase the popularities of high volume escalators (yellow circled regions), there is an increase in pedestrians nearby and also in the connecting corridors, which there is a decrease of pedestrians in more isolated regions. There is also a corresponding increase in the Pearson correlation coefficient, which makes sense as the nature of those escalators are changed from low volume to high volume (which is the case in the ground truth).

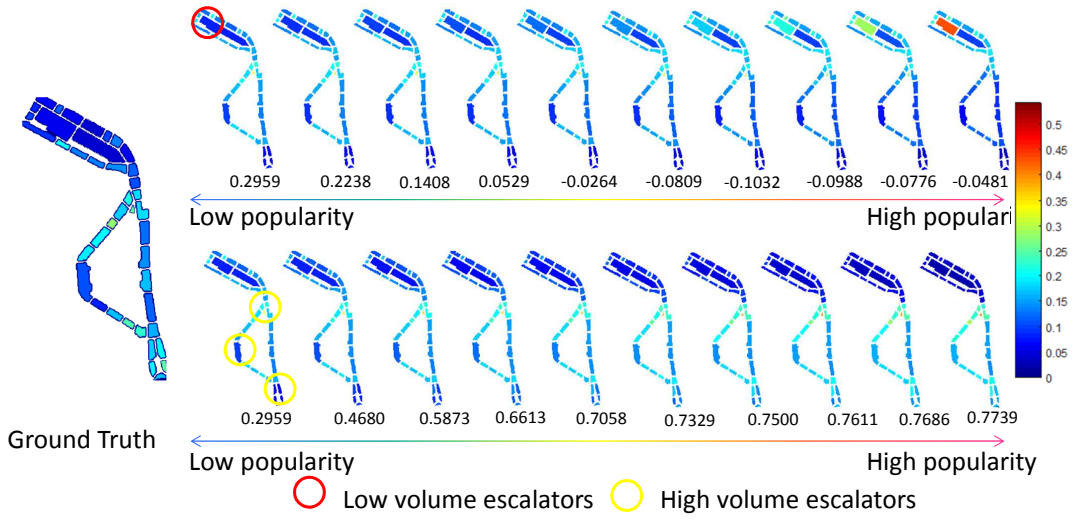


Figure 4.13: Visualization of $P(\zeta)$ as the popularities of non-area categories are varied for Ion floor B4. Top: varying the popularity of low volume escalators. Bottom: varying the popularity of high volume escalators. The numbers below the layouts are the Pearson correlation coefficients of the generated crowd density counts and ground truth crowd density counts (shown on the left).

4.8.3 Quantitative Results

Table 4.2 shows the log likelihood ratios and p-values obtained in our experiments. Column (a) shows the test results obtained using the f_γ learned from the training sets, in an LOOCV procedure. Column (b) shows the test results with in-test optimization in which the best f_γ's are found for each individual test floor. We also presented a combined log likelihood ratio by adding the log likelihood ratios for all the floors and a meta p-value by combining all p-values using Fisher's method as described in section 4.7.5.

For the results in column (a), we can see that most of the model predicted densities are more accurate than the densities predicted by the baseline distribution, except for Ion floors B2 and B3. Six of the nine better-than-baseline predictions (with negative log likelihood ratios) are statistically significant to 0.05 level, with another two significant to 0.1 level. The two worse-

than-baseline predictions are not statistically significant. Note that good density estimation results are obtained even though the training data was collected at different times over a 2.5 hour span, which suggest that the quasi-stationary assumption of the relative density distribution is reasonable.

For the results in column (b), we can see that all log likelihood ratios are negative (which means all model predictions are better than the baseline). Eight of the eleven predictions are statistically significant to 0.05 level, while the rest are significant to 0.1 level. In-test optimization is clearly helpful, but requires the architect to interactively set the popularities of non-area vessels. This shows that our model has sufficient degrees of freedom to correctly model the pedestrian density distributions, but it is unable to generalize the non-area vessel popularities from training data, likely because these popularities can only be modeled with inter-floor dependencies and external environment factors.

The two floors in column (a) that were poorly predicted by our model were Ion floors B2 and B3. The poor predictions were generally due to incorrect transfer of learned popularities of the non-area vessels in other floors, since the predictions significantly improved to be better than baseline with in-test optimization in column (b). We suspect that this incorrect transfer is due to special circumstances on these two floors that were not encountered on the other floors. For the case of Ion floor B2, there is a major mall entrance that connects to multiple destinations including the Wheelock Place shopping mall. This mall entrance has much higher traffic than other mall entrances on other floors, and therefore not previously observed in the training set. For the case of Ion floor B3, the high traffic mall entrance on Ion floor B2 also has an impact here, as it is close to an escalator that connects Ion floor B2 to B3. This means there is also a higher volume of traffic moving along this escalator here than is observed for other escalators in the training set. This inter-floor correlation is not represented in our model. In the future, we may want to investigate interconnections between floors that can be optimized jointly for

	(a)		(b)	
	Using interpolated path descriptors and learned f_γ		Additionally with in-test optimization of non-area popularities	
	Log likelihood ratio, t	p-value	Log likelihood ratio, t	p-value
Ion floor 1	-23.394889	0.001800	-35.530724	0.000600
Ion floor 2	-5.260449	0.018900	-5.996256	0.015400
Ion floor 3	-2.289187	0.070400	-2.871972	0.057400
Ion floor 4	-2.244838	0.077800	-2.640422	0.069200
Ion floor b1	-11.694704	0.010200	-21.316309	0.004100
Ion floor b2	1.989595	0.153900	-12.843013	0.047700
Ion floor b3	31.922641	0.123300	-12.411431	0.011800
Ion floor b4	-12.337599	0.017700	-31.349717	0.000900
Novena Square floor 1	-12.173411	0.004500	-21.833873	0.003700
Novena Square floor 2	-10.569334	0.008700	-13.367716	0.002000
Novena Square floor 3	-1.515239	0.117800	-1.678428	0.099800
Total threshold or combined p-value using Fisher's method	-47.5674	5.49434e-09	-161.84	1.83786e-12

The log likelihood ratio t compares the likelihood of generating ground truth density counts from the baseline uniform multinomial distribution to our model distribution. A negative t means the model distribution fits better.

Table 4.2: The log likelihood ratios and p-values obtained in our experiments. Column (a) shows the test results obtained using the f_γ learned from the training sets, in an LOOCV procedure. Column (b) shows the test results with in-test optimization in which the best f_γ's are found for each individual test floor.

an entire building, which is not possible currently due to the limited amount of data we have been able to collect.

4.8.4 Interpolation with Path Preference Priors

In this section, we discuss the results of experimenting with a range of weights for linearly interpolating learned path preference parameters with priors for these parameters. As described in section 4.3.3.1, the purpose of this regularization is to prevent overfitting.

We carried out the experiment by iterating through β values from 0 to 1 at an interval of 0.025, where small values of β mean that the priors dominate, while the converse is true for large values of β. At each β value, we conducted the LOOCV test as described in section 4.7.6, except that the learned path preference parameters are interpolated with the priors using the

β value. We computed a combined normalized log likelihood ratio:

$$S = \lambda_{\text{model}} - \lambda_{\text{baseline}} \tag{4.23}$$

where λ_{model} and $\lambda_{\text{baseline}}$ are normalized likelihoods of our model and baseline respectively, computed from (4.23). The normalized likelihoods are used as it would prevent floors with more people from dominating the sparser floors.

The normalized log likelihood ratios for all the iterated β values are plotted in figure 4.14. The log likelihood ratios form a U-shape curve with the minimum at $\beta = 0.2$ which means that a mixing of 20% learned path preference parameters and 80% priors is the optimal combination. Although the improvement from using interpolated parameters over using only priors is small relative to purely using learned path preference parameters, we also cannot rely only on priors and that 20% of learned path preference parameters is needed to shift the priors to get the optimal log likelihood. However it is also clear we cannot use the learned path preference parameters directly because we only have limited data for our experiments which may have led to overfitting and thus poorer results when testing.

As stated in section 4.8, we used this optimal value of $\beta = 0.2$ for the results previously presented in sections 4.8.1 and 4.8.3, and subsequently in later sections as well.

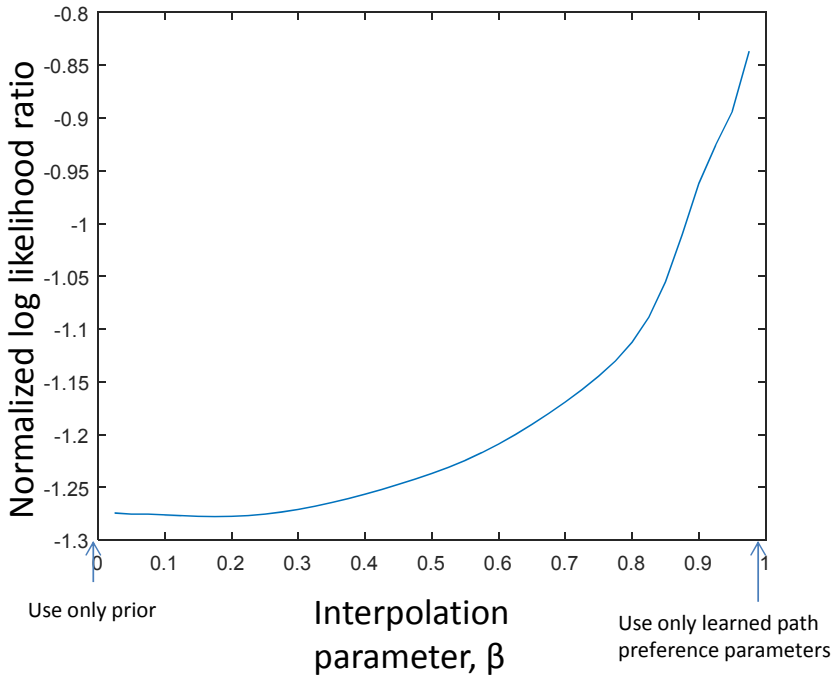


Figure 4.14: Log likelihood ratios when interpolating path preference parameters with varying amounts of priors.

4.8.5 Evaluation of Across-Mall Learning

Besides learning and testing across floors and malls using leave-one-out cross validation, we are also interested in finding out if the parameters (both for route preferences and category popularities) learned from one mall can successfully apply to another mall with a vastly different layout. This is a much more challenging situation due to the severe limitation of data collected.

In this experiment, we learn model parameters from all eight floors in the Ion mall and test the accuracy of the model prediction for each of the three floors in the Novena Square mall. We do not carry out the evaluation in the reverse direction because we only have data for three floors of the Novena Square mall which is too sparse and will lead to overfitting. Figure 4.15 illustrates learning from the Ion mall and testing on the Novena Square mall.

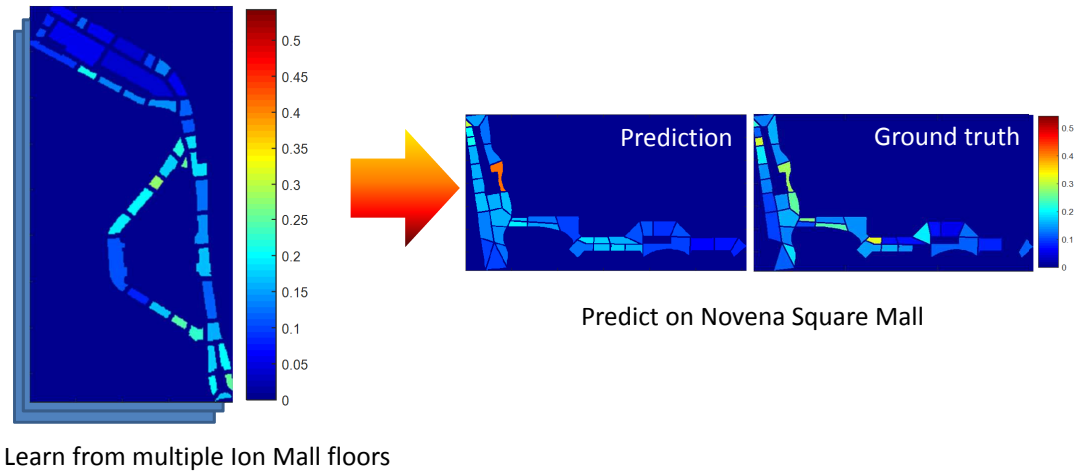


Figure 4.15: Prediction on the Novena Square mall using parameters learned from the Ion mall.

Table 4.3 shows the prediction results of learning from the Ion mall and testing on the Novena Square mall, with and without in-test optimization. The good results (low log likelihood ratios with small p-values) shows that transfer learning is indeed possible across malls, even with vastly different floor layouts. Future research with substantially more data from a greater variety of malls will allow us to be more conclusive about this.

4.8.6 Interpretation of Learned Path Preference Parameters

In this section, we will interpret the path preference parameters that are learned in our framework. For this purpose, we used a training set comprising all 11 floors from both malls for learning. We do so because there is some variation in the path preference parameters learned in each fold of the LOOCV procedure, and we want to do our analysis for the most general case.

Figure 4.16 shows how the sigmoid functions in (4.3) vary with different path descriptor values of expanse, line of sight, path distance and turn distance. In the graphs, we limit the plots to an x-axis range of -1 to 1. This is because the input path descriptor values are normalized

	(a)		(b)	
	Using interpolated path descriptors and learned f_γ		Additionally with in-test optimization of non-area popularities	
	Log likelihood ratio, t	p-value	Log likelihood ratio, t	p-value
Novena Square floor 1	-11.925746	0.004800	-19.189570	0.005200
Novena Square floor 2	-10.778636	0.008300	-11.701723	0.007000
Novena Square floor 3	-3.377098	0.037500	-3.422588	0.033200
Total threshold or combined p-value using fisher	-26.081480	0.000155947	-34.313881	0.000129866

The log likelihood ratio t compares the likelihood of generating ground truth density counts from the baseline uniform multinomial distribution to our model distribution. A negative t means the model distribution fits better.

Table 4.3: The log likelihood ratios and p-values obtained in our experiments in which we jointly learn from all floors in the Ion mall and test on each floor in the Novena Square mall. Column (a) shows the test results obtained using the f_γ learned from the training sets, in an LOOCV procedure. Column (b) shows the test results with in-test optimization in which the best f_γ's are found for each individual test floor.

by the means and standard deviations so that 68% of these values fall within the one standard deviation range. These sigmoid functions indicate the relative importance of each feature as the feature value increases; whether these function outputs are positively or negatively correlated, and for different ranges of feature values whether the rate of change is substantial or plateaued.

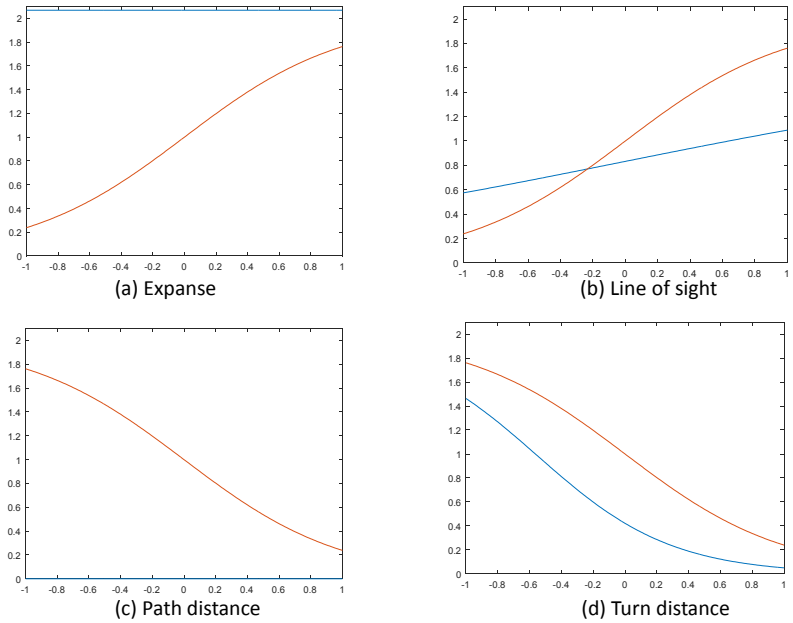


Figure 4.16: Sigmoids of $P(\eta|\vartheta_{in}, \vartheta_{out})$ in (4.2) based on estimated path preference parameters. Red lines are sigmoids based on prior path preference values, while blue lines are sigmoids with the optimal interpolation of learned and prior path preference values. The interpolated sigmoids for (a) and (c) are degenerate horizontal lines because their interpolated thresholds, c'_j in (4.7), remain far to the left or right of the graphs.

From figure 4.16(b), we can see that paths with greater line of sight are preferred and this fits our intuition that people prefer paths along which they can see further when walking. The path preference changes almost linearly with the average line of sight available while on a path. From figure 4.16(d), we can see that paths with smaller turn distances are preferred. This also fits our expectation that people prefer less winding paths with fewer turns. Unlike the line of sight descriptor, path preference does not change linearly with turn distance and the rate of change decreases for more extreme turn distances. This suggests to us that people are more sensitive to differences in turn distance when comparing paths that have close to the average turn distance, but when comparing two paths with either very large or very small turn distances, the turn distance descriptor does not play a major role in their choices of paths.

The horizontal plot in figure 4.16(c) suggests that pedestrians totally ignore path distances when choosing between paths, which is very surprising and counters to our intuition that people would prefer shorter paths. This may be due to the turn distance descriptor correlating strongly with the path distance descriptor on the floors in our training set, and thus when the preference parameters for these two descriptors are learned jointly, the model only chooses one of them to represent the shared characteristic. Figure 4.16(a) also suggests that pedestrians do not care about the expanse of paths when choosing between them. This is counter intuitive as we normally assume that pedestrians will prefer wider paths with bigger open spaces. Similar to the relationship between the path distance and turn distance descriptors, we expect that line of sight is likely correlated with expanse, hence the preference parameters for only one descriptor is significant, i.e. one of the path descriptors becomes unimportant when both descriptors are used together.

The algorithm may have learnt to accept most datapoints as it may has not enough datapoints to average out the noise. There is room for improvement in this part of the model.

4.8.7 Comparison to Agent-Based Pedestrian Simulation

In this section, we compare our method to the agent-based pedestrian simulation method [48] incorporating space syntax. In this method, a dense grid is superimposed on each floor layout such that there are on average 100 grid points in each node. Each agent will attempt to traverse from a source grid point to a sink grid point through a set of adjoining grid points, for which the source and sink grid points are prespecified. In order to determine the path connecting the source and sink grid points, at each time instance an agent randomly selects a grid point that is in its field of view and which is closer to the destination than its current position. A local trajectory is then determined by connnecting adjoining grid points between the selected grid point and its current position. Note that while this implicitly makes the agent prefers directions

that have greater line of sight, there are no external parameters to be set manually or learned from training data. In our experiments, we set the grid point at the center of each node as a potential source or sink. At each agent cycle, we randomly select a pair of source and sink grid points and ran the simulation to determine the grid points that the agent will traverse. After numerous cycles, we can compute the number of times that each grid point is visited by agents. The simulation cycles are continued until the relative accumulated densities between any two grid points remain relatively constant. After convergence, we estimate the density distribution at node level by summing up the densities of all grid points in each node.

In order to have an experiment in which path preferences are important, we used Ion floor B2 which has multiple loops. We computed the density distributions for this floor using both the described agent-based method and our framework.

It turns out that the simulated agents are much more unlikely to travel into the more remote regions, which can be indirectly observed in figure 4.17(b), where the pedestrian densities in remote regions are substantially under predicted, while more central regions are over predicted. Our model prediction in figure 4.17(a) is closer to the ground truth in (c). One likely difference is that the agents in the simulation continually make path decisions based on their immediate surroundings which are more applicable to pedestrians in a new environment, whereas in our model it is implicitly assumed that the pedestrians are acquainted with the environment and can conduct pre-trip path planning.

Next we want to compare probabilities given to different paths between a pair of origin and destination portals as estimated by the two frameworks. In figure 4.18, we computed the probabilities for three paths A,B and C, using both our model as well as the agent-based pedestrian simulation model. It may be observed that pedestrian simulation is overwhelmingly in favor of wider paths while our model has a much more even distribution of selected paths between the designated origin and destination portals. In addition, agent-based simulation is time consuming, particularly when attempting to provide sufficient coverage of all potential paths between

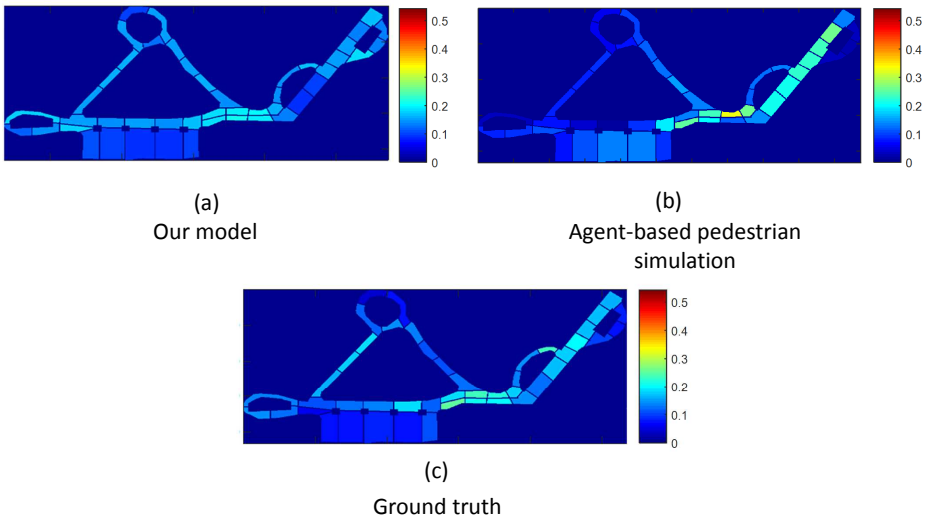
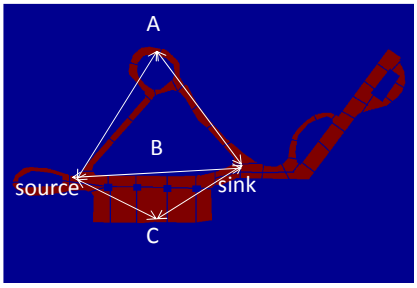


Figure 4.17: Comparing density prediction of agent-based pedestrian simulation and our model prediction. Pedestrian simulation agents are much more unlikely to travel into the more remote regions (see (b)), where the pedestrian densities in remote regions are substantially under predicted, while more central regions are over predicted. Our model prediction (a) is closer to the ground truth in (c).

all pairs of portals. For example, if we wanted to cater for a 90% chance of simulating the least probable path at least once in the floor layout of figure 4.17, we would theoretically need to run the simulation (written in Java by the authors of [48]) for 47 minutes. See appendix A.5 for derivation of this expected timing. This contrasts with our approach which currently takes 3 minutes in Matlab.

4.9 Summary

We have developed a probabilistic model to predict pedestrian density distributions in shopping malls based on route choice modeling. We are able to learn path preference parameters directly from observed pedestrian counts without the need to extensively track pedestrians. We implicitly model our pedestrian traffic as entering from or leaving area vessels (shops) and non-area vessels (mall entrances and escalators). In our framework, we model the pedestrian density



Our model
A:0.1219
B:0.4040
C:0.4741

Space syntax based pedestrian simulation
A:0
B:0.1717
C:0.8283

Figure 4.18: Comparing path probabilities of agent-based pedestrian simulation and our model between a fixed source sink.

distribution on a floor as influenced by path preferences and vessel popularities. The path descriptors used are expanse, line of sight, path distance and turn distance which are widely used in architecture theory. These model parameters are learned from ground truth pedestrian counts collected from existing malls. Based on these parameters, we can predict pedestrian density distributions better than a baseline uniform multinomial distribution, as demonstrated in our experimental results. We also show that compared to an agent-based pedestrian simulation, our model is both more accurate and much faster.

Chapter 5

Learning Category Popularities

5.1 Overview

In the previous chapter in section 4.3.6, we had assumed in our model that the popularities of area vessel categories, as represented by $P(\gamma)$, are proportional to the relative floor areas contained within corresponding shops. However, this is a gross simplification. Implicit in that model are a number of assumptions, including that the density of shoppers in each shop is the same, and the amount of time they spent within each shop is also the same.

In this chapter, we will provide an analysis of other factors influencing $P(\gamma)$, and propose a means of learning how the popularities of shop (i.e. area vessel) categories may vary beyond the areas of the shop floors.

5.2 The Physics of Popularity

In this thesis, the term "popularity" is not used from a psychological but rather a purely mathematical perspective. For a shop corresponding to the vessel ϑ, the popularity is simply the probability if a random pedestrian among those transiting between shops (inclusive of escalators and mall entrances) was picked at some random time instance, that she was heading

towards (or leaving) that shop, denoted by $P(\vartheta)$.

To probe further into this notion of popularity, consider the context of our problem. The assumption is that we are working on a timescale in which the pedestrian count on a floor is constant, and the pedestrian density distribution is stationary. A further derived assumption is that we are broadly dealing with a pervasive steady state scenario, in which the number of shoppers in every shop remains statistically constant. In other words, the flow of pedestrians into a shop, $\text{flow}(\vartheta)$ in say units of number of persons per minute, is constant and equal to the flow of pedestrians leaving the shop. If we were to consider all shopper flows into all shops, then we see that a randomly chosen pedestrian at a time instant will be heading to a shop with probability proportional to the flow, i.e.

$$P(\vartheta) \propto \text{flow}(\vartheta) \tag{5.1}$$

where the proportionality hides unknowns related to the total number of pedestrians and time constants dependent on the dimensions of the malls and walking speeds. We can consider shopper flow as a proxy for the vessel popularity $P(\vartheta)$.

Next, we investigate the components of flow. Intuitively, a simple model for shopping behavior is the process in which a shopper arrives at the shop, spends some amount of time within the shop, and then leaves. We call the average amount of time a shopper spends in the shop the *stay time*, denoted by $t_s(\vartheta)$.

Now if the stay time is dominated by time spent in a queue, the flow may then be approximated as

$$\text{flow}(\vartheta) = \frac{\kappa}{t_s(\vartheta)} \tag{5.2}$$

where κ is the number of parallel queues.

If on the other hand the stay time is dominated by activities that are carried out in parallel by different shoppers, for example browsing for clothes in an apparel store or dining in a

restaurant, the limiting factor is likely the typical customer density in the store. We can denote this density as $d_p(\vartheta)$, in say units of persons per squared meter. Here the pedestrian flow into the shop may be approximated as

$$\text{flow}(\vartheta) = \frac{d_p(\vartheta)}{t_s(\vartheta)} \text{area}(\vartheta) \tag{5.3}$$

where $\text{area}(\vartheta)$ gives the internal floor area of the shop ϑ.

5.3 Invariant Factors in Area Vessel Flow

In the previous chapter we had assumed that the popularity and thus pedestrian flow is proportional to the area of the shop floor, which meant that the term d_p/t_s was treated as invariant across different shops.

A deeper examination would reveal that this is unlikely to be true. Consider the following examples. The typical person-to-person distance in a supermarket is significantly larger than in a food court (imagine trying to navigate a trolley in a food court!), and thus d_p should be smaller for the supermarket. A trip to the pharmacy is likely much shorter than a visit to the hairdresser, and hence t_s will be much smaller for the pharmacy. Although in some cases the differences in d_p and t_s may cancel out in the division, this is very unlikely to be widely true. See figure 5.1.

However, it may be true that d_p and t_s are invariant within *categories* of shops, even though they are different between categories. Hence we propose that with an appropriate categorization of shops, that the term d_p/t_s is constant within each category. We call this term the *flow density*, and may be expressed as

$$q_\gamma = \frac{d_p(\gamma)}{t_s(\gamma)} \tag{5.4}$$

where $d_p(\gamma)$ and $t_s(\gamma)$ are the category-dependent versions of the customer density and stay time respectively.

(a) Jewelry Shop

(b) Food Court

(c) Convenience Store

(d) Hair Salon

Figure 5.1: Informal comparison of four shops in different categories. A food court (b) is often more crowded and has smaller person-to-person distances than a jewelry shop (a) which is often quite empty; the food court therefore has higher customer densities d_p. The stay time t_s in a hair salon (d), typically 30 to 60 minutes, is usually much longer than in a convenience store (c), where customers typically go to buy one or two items quickly.

The idea is then to learn the different q_γ for different area vessel categories from labeled training data. This will allow a more nuanced prediction of the pedestrian density distribution in a new mall layout, for which a suggested shop space zoning / categorization has been given. There is though still the problem of determining the best way to categorize different shops.

5.4 Categorization of Vessels based on Flow Densities

When we look up a typical mall directory, we see shops classified into conventional semantic categories such as food, beauty & wellness, jewelry & watches, fashion & accessories, shoes &

(a) Burger King (Fast Food)

(b) Tcc (Cafe)

Figure 5.2: An example comparison: Burger King and Tcc have different flow densities, measured at 0.35 and $0.05 pax/min/25m^2$ respectively. This is due to different stay times (customers tend to linger longer in cafes) and customer densities (fast food restaurants are typically more crowded).

bags and sports. Although these categories are useful for customer search, they are too broad for consistency of flow densities. For example in the food category, a fast food restaurant such as Burger King has a flow density very different to that of a cafe such as Tcc, because the typical stay times and customer densities are different (see figure 5.2). In fact, when we manually measured and computed the flow densities, they turned out to be 0.34 and 0.05 respectively in units of persons per minute per 25 squared meters (of shop area).

For consistency of flow densities within categories, we subdivided the semantic categories into smaller categories within each shop has the same flow densities. The food category is subdivided into snacks, food courts, restaurants, cafes and fast food. Snack stores are mainly for takeaway and therefore have the lowest stay times in the food category. Restaurants have longer stay times because the customers are seated and have to wait for the food to be prepared and served. Food courts also have seats but customers stay much shorter times. The beauty and wellness category is subdivided into personal care, cosmetics, hairdressing and pharmacies, as they have different attraction rates and price ranges, and thus different flow densities. For the jewelry & watches category, the price range in jewelry shops is very different to that in watch shops, and this has a significant impact on the popularities of these shops, with jewelry shops

having much lower flow densities than watch shops. We therefore treat jewelry and watch categories separately. We also divided the fashion & accessories category for similar reasons. For the remaining semantic categories such as gifts, sports and shoes & bags, we retain those categories as we consider the shops within these categories to have consistent flow densities. The complete list of shop categories used in our framework has already been presented in table 4.1.

5.5 Learning Flow Densities of Area Vessel Categories

We can now establish a model for the popularity of an area vessel category γ:

$$P(\gamma) \propto \text{flow}(\gamma) = q_\gamma \text{area}(\gamma) \tag{5.5}$$

For clarity of notation, we will consider each vessel category to have a q_γ, regardless of whether it is an area or non-area vessel category. For each non-area vessel category, we may relate the f_γ in (4.14) to q_γ by assuming that the category has unit floor area. We may then change (4.14) into

$$P(\gamma) = \begin{cases} \frac{1}{X} q_\gamma \sum\limits_{\vartheta_k \in \mathscr{V}_\gamma} \text{area}(\vartheta_k) & , \gamma \text{ is an area vessel category} \\ \frac{1}{X} q_\gamma & , \text{otherwise} \end{cases} \tag{5.6}$$

where

$$X = \sum_{\gamma \notin \mathscr{C}_n} q_\gamma \sum_{\vartheta_k \in \mathscr{V}_\gamma} \text{area}(\vartheta_k) + \sum_{\gamma \in \mathscr{C}_n} q_\gamma \tag{5.7}$$

and subject to

$$\begin{aligned} \sum_\gamma q_\gamma &= 1 \\ q_\gamma &\geq 0. \end{aligned} \tag{5.8}$$

As before, $\mathscr{V}_\gamma$ is the set of all vessels in vessel category γ, $\text{area}(\vartheta_k)$ is the area of vessel ϑ_k, and $\mathscr{C}_n$ is the set of all non-area vessel categories. The constraint that all q_γ's sum to one is

needed to prevent multiple solutions of q_γ that will lead to the same optimal point due to the normalizing factor X.

The learning framework is similar to that in chapter 4, except that we want to find optimal flow densities q_γ for all vessels, instead of only f_γ for non-area vessels in (4.14). The optimization is carried out by minimizing the negated joint log likelihood function (4.23) with respect to path preference parameters **a**, **b** and **c**, as well as all q_γ's. This is carried out by interleaved optimization and repeated until convergence, in a similar manner to the previous chapter. For detailed steps of the algorithm, see appendix A.6.

5.5.1 Prediction Results for Pedestrian Density Distribution

The numerical results of leave-one-out cross validation are shown in table 5.1. The left half (a) are the results from a conventional testing procedure, while the right half (b) are results after additional in-test optimization. For the results in column (a), six of the ten better-than-baseline predictions are statistically significant to 0.05 level, with another three significant to 0.1 level. However there is an unusual result for Novena Square floor 1 in part (a), in which the model prediction is much poorer than the baseline, and yet the p-value is very small. Further analysis reveals that our model prediction appears similar to a delta function, with a single large peak in one of the nodes. Such delta function distributions will lead to very low likelihood values for samples that are not drawn from those exact distributions, which explains the very high log likelihood ratio. However, because the ground truth pedestrian counts is somewhat correlated with the model prediction, it still leads to a higher likelihood than other randomly sampled counts, which explains the small p-value. For the results in column (b), all model predictions are better than the baseline, with eight of the eleven predictions statistically significant to 0.05 level, and the remaining three significant to 0.1 level.

The log likelihood ratios are on average similar to those in table 4.2 in the previous chapter. Overall, it seems that learning category-specific flow densities did not help in improving overall

	(a)		(b)	
	Using interpolated path descriptors and learned q_γ		Additionally with in-test optimization of non-area q_γ	
	Log likelihood ratio, t	p-value	Log likelihood ratio, t	p-value
Ion floor 1	-32.429190	0.002100	-33.830685	0.000900
Ion floor 2	-5.254653	0.024500	-5.272329	0.023000
Ion floor 3	-0.435356	0.188000	-2.292198	0.071900
Ion floor 4	-0.737920	0.096800	-1.978127	0.063900
Ion floor b1	-22.408972	0.002900	-23.439413	0.002400
Ion floor b2	-6.063741	0.084100	-12.463490	0.048500
Ion floor b3	-2.125970	0.035100	-13.098092	0.009300
Ion floor b4	-2.898345	0.042300	-31.469752	0.001500
Novena Square floor 1	28.281026	0.005500	-21.413973	0.003900
Novena Square floor 2	-12.790671	0.001700	-13.595682	0.001500
Novena Square floor 3	-2.163778	0.068600	-2.372004	0.052300
Total threshold or combined p-value using Fisher's method	-59.027569	1.66285e-09	-161.225744	2.07193e-12

The log likelihood ratio t compares the likelihood of generating ground truth density counts from the baseline uniform multinomial distribution to our model distribution. A negative t means the model distribution fits better.

Table 5.1: Prediction thresholds (maximum likelihood ratios) using learned q_γ (learned flow densities) and their p-values.

prediction performance compared to using a purely area-based model in the previous chapter. We further investigate whether the learned flow densities and inferred flows are related to actual flows in the next sections.

5.5.2 Data Collection of Actual Customer Flows

To get insights as to whether the learned flow densities reflect reality, we subsequently collected additional data on the frequencies of customers entering and leaving shops for two of the floors. We captured 5-minute video clips of every shop entrance on the B2 floor of the Ion mall and the first floor of the Novena Square mall. For the Ion shopping mall, the videos were taken on Sunday 5 October 2014 from 4:14pm to 6:45pm, while for the Novena Square shopping mall on Saturday 11 October 2014 from 3:15pm to 6:00pm. It was subsequently realized that videos were missed for all the non-area portals, i.e. escalators and entrances, in Ion, while the video for a single entrance was missed for Novena Square. To rectify this, we captured videos for the missed portals on Saturday 18 October 2014, with the Novena Square entrance captured at

5:05pm to 5:09pm, and the Ion missing portals from 5:20pm to 6:38pm.

We then counted the number of people entering and leaving these portals in the video clips, and combined the counts for portals associated with the same vessels. We then computed the rate of customers entering each vessel and also the rate of customers leaving each vessel in terms of number of persons per minute. We then specified the customer flow of each vessel to be the average of these two rates. Tables 5.2 and 5.3 show all the customer flows, separately for each of the two floors.

In this chapter, we do not analyze customer flows from non-area vessels such as escalators and mall entrances. The reason for not doing so is based on earlier results, it seems that we do not have a viable model for predicting customer flows for non-area vessels. From (4.14), the flows f_γ are assumed to be invariant within the same non-area vessel categories, because there are no corresponding areas to allow for the concept of invariant flow densities to be used in (5.3) for modeling the flows of area vessel categories. From the results of our in-test optimization experiments in section 4.8.2, in-test optimized flows of non-area vessel categories significantly vary from floor to floor and therefore not invariant as previously assumed. This model for predicting non-area vessel category flows is too simple to be accurate. In the future, we may improve our model to include more factors for non-area vessel category flows such as proximity to other shopping malls and carparks.

It can be informally observed that shops such as the H&M clothing store and the Cold Storage supermarket have high customer flows. This matches our intuition that larger shops have higher popularities, i.e. area correlates with flow. In the next section, we will verify the similarities of values of our learned vessel category popularities $P(\gamma)$ with measured vessel flows aggregated at category level.

Shop Name	Vessel Category	Area(m²)	In	Out	Mean in and out	Duration (sec)	Measured Flow/min
Kase	Accessories	35.14	5	6	5.5	300	1.1
Accessorize	Accessories	102.70	10	9	9.5	222	2.567567568
L'Occitane	Personal care	39.20	5	5	5	300	1
The body shop	Personal care	59.60	12	9	10.5	300	2.1
Tcc	Cafés	121.00	9	2	5.5	480	0.6875
Yvessaintlaurent	Cosmetics	33.59	7	3	5	300	1
Victoria secret	Cosmetics	218.44	1	0	0.5	316	0.094936709
Estee lauder	Cosmetics	47.12	3	5	4	300	0.8
Chanel	Cosmetics	45.02	3	1	2	300	0.4
Dior	Cosmetics	46.57	0	1	0.5	300	0.1
Shu eumera	Cosmetics	54.14	3	3	3	300	0.6
Uniqlo	Fashion	660.18	44	26	35	300	7
H&m	Fashion	919.11	83	132	107.5	321	20.09345794
Levi's	Fashion	208.49	4	3	3.5	300	0.7
Factorie	Fashion	252.17	25	25	25	360	4.166666667
Mango	Fashion	381.88	11	8	9.5	300	1.9
Staradivarius	Fashion	307.09	50	27	38.5	300	7.7
Topshop	Fashion	722.60	11	20	15.5	300	3.1
Zara accessories	Fashion	732.19	46	43	44.5	300	8.9
Promod	Fashion	131.62	10	9	9.5	300	1.9
Pull and bear	Fashion	345.97	26	40	33	300	6.6
La senza	Fashion	182.61	0	0	0	300	0
Bershka	Fashion	393.34	53	52	52.5	348	9.051724138
Steve madden	Shoes / bags	70.85	13	13	13	300	2.6
Clark's shoes	Shoes / bags	94.96	5	3	4	300	0.8
Ecco	Shoes / bags	104.73	5	3	4	300	0.8
Skechers	Shoes / bags	109.03	13	17	15	300	3
Geox	Shoes / bags	75.50	0	0	0	300	0
Aldo	Shoes / bags	127.04	18	29	23.5	300	4.7
Cath kidston	Shoes / bags	52.29	3	5	4	300	0.8
Fossil	Shoes / bags	137.54	7	14	10.5	300	2.1
Puma	Sports	107.67	10	20	15	300	3
Kikki.k	Stationery	101.83	8	12	10	318	1.886792453
Swatch	Watches	40.43	7	2	4.5	300	0.9
Moments of city chain	Watches	78.64	11	11	11	300	2.2

Table 5.2: Collected customer flow data of every shop in Ion floor B2. Columns 1 (Shop Name) shows the name of each shop vessel. Column 2 (Vessel Category) shows the vessel category of each vessel. Column 3 (Area) shows the area of each shop vessel. Column 4 (In) shows the number of people moving into each shop vessel over the period of time specified in column 7 (Duration) and likewise column 5 (Out) shows the number of people moving out of each shop vessel over the same period of time. Column 6 (Mean in and out) is the mean of column 4 (In) and column 5 (Out). Column 8 (Measured Flow/min) is the mean number of people moving in and out of each shop per minute.

Shop Name	Vessel Category	Area(m^2)	In	Out	Mean in and out	Duration (sec)	Measured Flow/min
Lovisa	Accessories	31.81	3	1	2	300	0.4
Helen	Accessories	31.09	7	5	6	300	1.2
Oub	Banks	323.71	0	0	0	300	0
Body shop	Personal care	53.53	0	0	0	300	0
Hans	Cafés	108.32	13	14	13.5	300	2.7
Cedele	Cafés	83.54	6	4	5	300	1
Baskin robin	Cafés	49.09	0	3	1.5	300	0.3
The soup spoon	Cafés	55.99	5	0	2.5	300	0.5
Spinelli	Cafés	38.89	0	0	0	300	0
Harrys	Cafés	73.82	0	0	0	300	0
Tcc	Cafés	95.24	2	0	1	300	0.2
Ramen monster	Cafés	46.57	6	0	3	300	0.6
Sasa	Cosmetics	54.91	3	6	4.5	300	0.9
Home fix	DIY stores	83.78	1	1	1	300	0.2
I studio	Electronics	72.98	23	17	20	300	4
Cotton on	Fashion	183.88	1	8	4.5	300	0.9
Espirit	Fashion	67.45	16	15	15.5	300	3.1
Levi's	Fashion	71.24	0	0	0	300	0
KFC	Fast food	129.15	6	3	4.5	300	0.9
Burger King	Fast food	153.33	7	14	10.5	300	2.1
Mos burger	Fast food	70.82	2	2	2	300	0.4
Ec house	Hairdressers / barbers	11.34	1	1	1	300	0.2
Gnc	Health supplements	32.95	0	2	1	300	0.2
Feriaia	Chocolates / wines / flowers	13.98	1	1	1	300	0.2
Ramen play	Restaurants	37.99	0	2	1	300	0.2
Charles & Keith	Shoes / bags	68.05	3	4	3.5	300	0.7
Cha time	Snacks	21.36	2	3	2.5	300	0.5
Old chang kee	Snacks	12.24	4	4	4	300	0.8
Breadtalk	Snacks	73.22	1	7	4	300	0.8
Sf	Snacks	18.00	7	7	7	300	1.4
Columbia sportswear company	Sports	63.19	3	2	2.5	300	0.5
Salomon	Sports	77.12	3	6	4.5	300	0.9
Running lab	Sports	95.30	11	10	10.5	300	2.1
Sportiv 360	Sports	123.15	9	2	5.5	300	1.1
Urban 360	Sports	120.03	7	0	3.5	300	0.7
Sun paradise	Sports	51.79	3	1	2	300	0.4
New balance	Sports	110.78	8	3	5.5	300	1.1
Guardian	Pharmacies	118.71	1	5	3	300	0.6
Cold storage	Supermarkets	1272.82	42	39	40.5	300	8.1

Table 5.3: Collected customer flow data of every shop in Novena Square floor 1. Columns 1 (Shop Name) shows the name of each shop vessel. Column 2 (Vessel Category) shows the vessel category of each vessel. Column 3 (Area) shows the area of each shop vessel. Column 4 (In) shows the number of people moving into each shop vessel over the period of time specified in column 7 (Duration) and likewise column 5 (Out) shows the number of people moving out of each shop vessel over the same period of time. Column 6 (Mean in and out) is the mean of column 4 (In) and column 5 (Out). Column 8 (Measured Flow/min) is the mean number of people moving in and out of each shop per minute.

5.5.3 Comparison of Model-based Category Popularities to Measured Flows

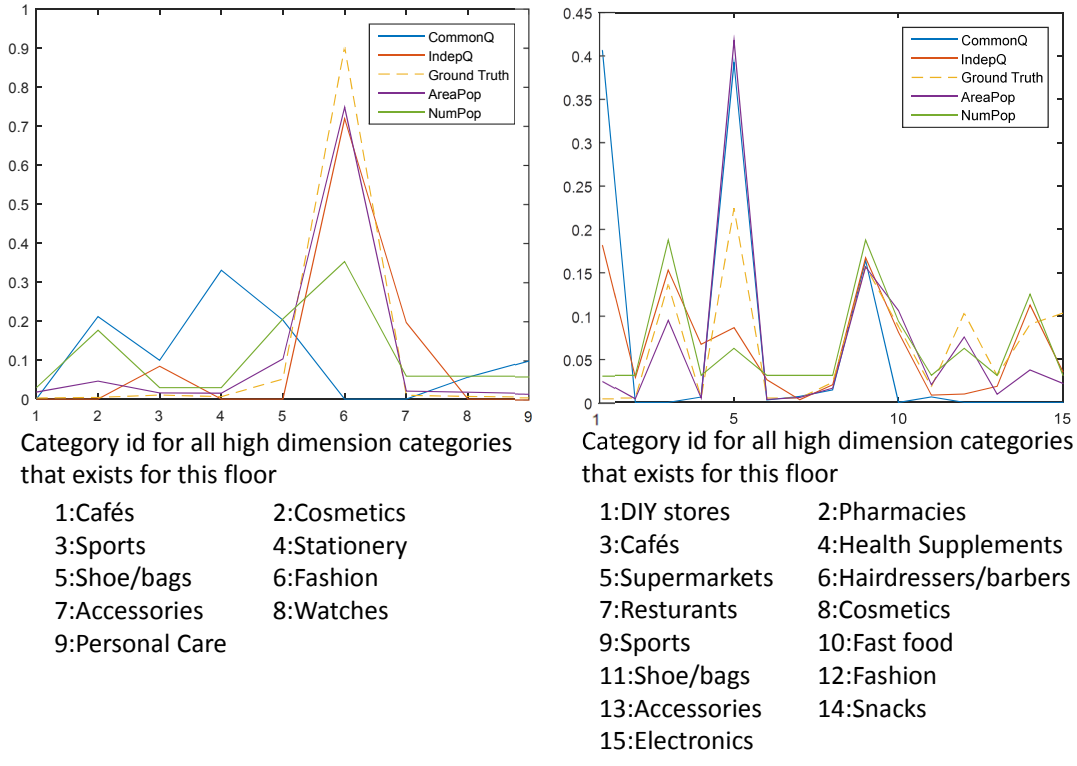
In this section, we will compare four models based on (5.6) for predicting shop vessel category popularities $P(\gamma)$ on Ion floor B2 and Novena Square floor 1, namely:

- **CommonQ**: Popularities inferred for a floor based on learning a common q_γ from all other floors,
- **IndepQ**: Popularities inferred for a floor based only on ground truth pedestrian counts for that particular floor, in which each floor may be considered to have independent q_γ's,
- **AreaPop**: Popularities defined as proportional to total floor area of all shop vessels in each category, i.e. q_γ is the same for all categories, and
- **NumPop**: Popularities defined as proportional to number of shops in each category, i.e. q_γ is constant as previously and the area of each vessel is set to one.

These four models of popularities are compared to the ground truth $P(\gamma)$ estimated from the customer flows obtained in section 5.5.2. Based on (5.1), the ground truth $P(\gamma)$ is simply obtained by normalizing the total customer flow for area vessels to unity. This is technically a conditional distribution: if we are dealing with the probabilities of source vessel categories $P(\gamma_{out})$, we assume we know that the pedestrian in question is leaving from an area vessel, i.e. a shop; likewise for $P(\gamma_{in})$. Note that there is no restriction that the pedestrian is only moving between area vessels. Similarly, we apply this conditional assumption in the analysis of the four $P(\gamma)$ models and exclude non-area vessel categories by setting $q_\gamma = 0$ for such categories after learning is completed and re-normalizing $P(\gamma)$ via (5.6).

Figure 5.3 shows the relationship between the $P(\gamma)$ values estimated from flow measurement, and those inferred by the various models listed above. For Ion floor B2 (figure 5.3(a)), the $P(\gamma)$ estimated by both the IndepQ and AreaPop models exhibit strong correlation with the ground

truth. Note that this was achieved despite the models using density data that is 1.5 months older than the ground truth, indicating that our assumption that relative density distributions remain steady over time is reasonable. The NumPop model has slightly weaker correlation, while the CommonQ model fits the ground truth poorly. The NumPop model overestimated the cosmetics category because while there is a large number of cosmetic shops, they do not typically have a large number of customers. Conversely, it underestimated the fashion category which is dominated by a single large and popular store, namely H&M. The CommonQ model performed surprisingly poorly, overestimating the cosmetics and stationery categories while predicting the fashion category to have zero popularity. This may reflect the difficulty of estimating latent parameters with limited data and even more so when attempting to transfer them to test cases.



(a) Ion floor B2

(b) Novena Square floor 1

Figure 5.3: Graphical comparison of four category popularity models with ground truth for Ion floor B2 and Novena Square floor 1.

For Novena Square floor 1 (figure 5.3(b)), all the models appear to have relatively good correlation with the ground truth in roughly 12 of the 15 shop categories. However, there was a number of mispredictions. The IndepQ model heavily overpredicted the DIY category, while underpredicting the supermarket category. This is likely due to the single DIY store located next to the single supermarket, and the learning framework had excessively apportioned the high pedestrian traffic to the DIY store. (Look at the section 5.5.3.1for more detailed analysis.) For the CommonQ model, there is a data insufficiency problem as there is only one other DIY store and supermarket in the training data. This led to overprediction of both the DIY and supermarket stores. The mispredictions of the supermarket category by the AreaPop

	Total Variation Distance against Ground Truth	
	Ion floor B2	Novena Square floor 1
CommonQ	1.8309	1.1585
IndepQ	0.5206	0.6644
AreaPop	0.3081	0.4704
NumPop	1.1003	0.5483

Table 5.4: Total variation distances between four category popularity models and ground truth for Ion floor B2 and Novena Square floor 1.

and NumPop models are somewhat expected as the customer densities within supermarkets are quite different from other shops and cannot be modeled solely by floor area or the number of shops.

In table 5.4, we compared the various model distributions $P_{\text{model}}(\gamma)$ to the ground truth $P_{GT}(\gamma)$ in terms of total variation distances. The total variation distance is calculated by

$$\sum_i |P_{\text{model}}(\gamma_i) - P_{GT}(\gamma_i)|. \tag{5.9}$$

Here a value of 0 means a perfect match between the two distributions, while larger values reflect increasing mismatch between the two distributions. From the table, we can see that the AreaPop distributions are closest to the ground truth for both floors. The IndepQ and NumPop distributions are ranked second and third in accuracy for Ion floor B2, and swap positions for Novena Square floor 1. The CommonQ distributions have the worst match to the ground truth.

It should be obvious that area-based popularity modeling only applies to area vessel categories. For non-area vessel categories such as escalators and mall entrances, learning the popularities from data leads to significantly better pedestrian density estimates than using arbitrary popularity values.

5.5.3.1 Failure Modes of Our Category Popularities model

In this section, we will look at the failure modes when attempting to learn category popularities using the IndepQ and AreaPop models as compared to ground truth popularities of shoppers entering and leaving shops. For Ion floor B2, our model accurately predicted the popularities of most categories except the Sports and Accessories categories. For Novena Square floor 1, popularities for ten of the shop categories were accurately predicted except the DIY, Health Supplements, Supermarket, Fashion and Electronics categories. The poorly predicted cases are analyzed below.

In (4.1) which models our overall framework, the observables are the nodes $P(\zeta)$ on the left hand side of the equation. Our learning algorithm aims to deconvolve the terms from left to right to get the paths $P(\eta)$, portals $P(\theta)$, vessels $P(\vartheta)$ and lastly the vessel category popularities $P(\gamma)$.

One scenario that leads to poor prediction involves certain "shared nodes", for which a single node may be connected to multiple portals. This is particularly problematic when these portals belong to single vessel categories (i.e. categories with only a single shop each). Under such conditions, the sole shops for these categories must all share the same paths, since they are all connected to the same node. For example if there are two such categories γ_1 and γ_2, then these categories must share the exact same set of paths η. This means we cannot distinguish between these categories during learning (without prior information about their differences), which can be argued from label permutation symmetry — switching the names of the categories (γ_1 to γ_2 and γ_2 to γ_1) will have no impact on the estimation of the popularities.

This scenario is encountered on Novena Square floor 1. There is a shared node near the left side of the floor layout which connects five portals belonging to the DIY, Florist, Health Supplements and Supermarket categories, and on this floor the first four categories are single vessel

categories, while there are two vessels in the Supermarket category[1]. However since the two supermarket vessels are connected to the same node and no other nodes, this makes the Supermarket category indistinguishable from a single vessel category. As expected, the estimated popularities for these categories have significant deviations from the ground truth. In fact the differences between the estimated category popularities are smaller than those in the ground truth, which illustrate the difficulty in distinguishing the categories. However the estimated popularities are not identical, likely due to numerical differences in optimization because the floor areas of the categories are different and thus the estimated flow densities q_γ are also different.

A weaker version of the scenario is encountered in Ion floor B2. Although there are no shared nodes with single vessel categories, there is a particular shared node near the center-right part of the layout connecting two portals, one each from the Accessories and Fashion categories. While there are many vessels in the Fashion category on this floor, there are only two vessels in the Accessories category. Although it is theoretically possible to determine these category popularities, the small number of vessels in the Accessories category meant the estimation is ill-posed. In particular for the shared node, the shop connected to it in the Fashion category is the H&M shop which has very high actual popularity, and we suspect that the estimated popularity has spilled over into the connected Kase shop in the Accessories category, which has much lower actual popularity. As a result, it turned out that the Accessories category popularity was overpredicted and the Fashion category popularity was underpredicted.

Another problem that affects shared nodes is that our framework does not model the movement of shoppers between portals sharing a node, e.g. a shopper who walks out of one shop and immediately enters a neighboring shop, where the entrances are co-located within the same node. Such movement could have been modeled using single node paths that start and end at

[1]In subsequent investigations, it turned out that a pharmacy was mislabeled as a supermarket in the ground truth data.

the same node and are one node long. However we do not yet have an appropriate way to model the path preferences for these paths, because these preferences depend on the line of sight path descriptor, and the line of sight computation depends on at least two nodes to determine the walking direction.

Besides the problem involving shared nodes, our analysis also reveals that in certain situations the invariant flow density assumption made in our framework is significantly violated. We had assumed that the flow density, as defined in (5.4), is invariant among shops within each area vessel category.

To investigate this further, we considered each multi-vessel category in Novena Square floor 1 and determined the amount of variation that exists in the flow densities of the shops. Based on (5.3), we can compute the flow density of a shop by dividing the mean flow of shoppers by the floor area of the shop. One way to characterize the amount of variation is to compute the standard deviation of the flow densities within a category. However, it turns out that the standard deviation depends on the magnitude of the flow density as follows. If we assumed that the flow of shoppers entering and leaving a shop roughly fits a Poisson distribution with some rate parameter λ (equivalent to the mean flow), then the distribution of λ across different shops in the category can be modeled with its conjugate prior, a gamma distribution. Without making assumptions about the parameters of the gamma distribution, we can state that the variance σ^2 of λ is roughly the 1.5th power of the mean μ of λ. Hence to compare the intra-category variation of flow densities *across* different categories, we computed for each category a normalized variation value

$$V_\gamma = \frac{\sigma_\gamma^{\frac{4}{3}}}{\mu_\gamma} \tag{5.10}$$

The multi-vessel categories ordered by ascending V_γ values (included in parentheses) are: Fast Food (0.234), Sports (0.368), Cafe (0.596), Snack (0.676) and Fashion (1.28), excluding the

Supermarket category which is affected by the abovementioned shared node problem. This shows that the flow densities of the shops within the Fast Food category is very consistent to each other, while there is high variation in the flow densities within the Fashion category, with other categories lying in between. In comparing these findings to the results presented in figure 5.3(b), we see that the ordering of these categories by V_γ values is exactly the same as the ordering of these categories by descending accuracies of the IndepQ model — the Fast Food, Sports and Cafe categories are very well predicted, the Snack category is less well predicted, while the Fashion category is poorly predicted.

Although the AreaPop model led to better prediction results, the analysis above also invalidates the AreaPop assumption that the flow densities across all categories are the same. The mean flow densities for the various multi-vessel categories are 0.209, 0.424, 0.219, 1.11, 0.244 and 0.143, and we can see that these are quite different from each other. The reason why the AreaPop model outperformed the IndepQ model is likely due to the difficulty of estimating a model with many more parameters using only limited data.

There also remains some cases for which we cannot find systematic reasons for poor prediction. However, the differences between these predictions and the ground truth category popularities are smaller as compared to the cases covered above. These categories include the Sports category on Ion floor B2 and Electronics category on Novena Square floor 1, both of which are single vessel categories. We attribute these differences due to random variations and transient noise when sampling the density count values.

5.5.4 Discussion

The AreaPop model most accurately models area vessel category popularities. This model assumes that the flow density q_γ in (5.6) is constant for each shop and each category. If we analyze the data in tables 5.2 and 5.3, we see that the ground truth flow densities vary from shop

to shop, ranging from 0.1 to 4.75 times of the unit q_γ assumed in the AreaPop model. This is as expected based on our discussion in section 5.3. However it turns out that the flow densities estimated from the IndepQ model vary from 0 to 14 times of the unit q_γ, while the CommonQ model estimates flow densities that vary from 0 to 20 times. These variations are much larger than those found in the ground truth data, and are likely due to the challenge of estimating a large number of latent variables from limited observations. Hence although the AreaPop model assumption of constant flow density is incorrect, it is nonetheless more accurate than the grossly varying flow densities estimated by the IndepQ and CommonQ models. Consequently, the NumPop model also estimates the popularities better than the IndepQ and CommonQ models, but not as well as the AreaPop model.

In comparing the two models that attempt to learn latent flow densities from pedestrian counts, we see that using the IndepQ model leads to closer estimates of the ground truth popularities than the CommonQ model. Since the IndepQ model is attempting to estimate model parameters to fit the pedestrian counts for a specific floor, as opposed to the CommonQ model in which model parameters are learned from pedestrian counts of other floors, it is natural to expect it to fit the pedestrian counts for the test floor much more accurately. This in itself does not explain why it will fit the latent popularities better. However, we suspect that the shop popularities are well correlated with the pedestrian counts directly in front of the shop entrances, and thus a more accurate estimate of the pedestrian counts will also lead to a better estimate of the shop popularities. The correlation arises when the dispersion directions of pedestrians entering or leaving each portal are uniformly distributed, which will not be entirely true but is perhaps a relatively reasonable approximation. Hence the more accurate pedestrian count prediction of the IndepQ model leads to better popularity estimates than the CommonQ model.

We had attempted to improve the accuracy of pedestrian count prediction by generalizing the previous model, in which all shop categories had the same flow density, to one in which different shop categories had flow densities that were learned from previous observations. This

did not turn out fully as intended as we did not observe an improvement in pedestrian count prediction when using the CommonQ model as compared to the previous AreaPop model. Furthermore, the learned latent flow densities from the CommonQ and IndepQ models turned out to be less accurate than those from the AreaPop and NumPop models. This is likely due to the large number of latent and model parameters, when coupled with limited observations, that resulted in over-fitting of the training data, and hence no additional improvement in the performance on test data. The most viable solution is likely to be an extensive data collection effort, which may be carried out in the future.

5.6 Summary

We have provided a framework for conceptualizing in a steady state scenario the flows of customers in and out of shops in terms of customer densities and stay times. The flows can be used as a proxy for the popularities of different shop categories as represented by $P(\gamma)$. We have postulated that the customer flow density is category specific but invariant within each category, and attempted to learn these flow densities as latent variables based on observed pedestrian counts. Based on actual measured flows, it turned out that a universally (rather than category specific) invariant flow density produced the best results with strong correlation to the measured flows, indicating that the popularities of different vessel categories are best modeled by simply using their corresponding shop floor areas. However, we suspect that this is likely due to limited observed training data, and more future experiments with a more extensive dataset will be needed for more definite conclusions. Due to time constraints in our work, we concentrate more on modeling the behaviors of the pedestrians than validating it. In the future, we may look into using simulation to analyze our model.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis had been focused on pedestrian detection and density modeling.

In chapter 3, we presented a novel second order intensity feature, based on image intensity Hessians, for representing image surfaces. We found that the Hessian feature detects similar image patterns to HOG when used alone, but it is able to adapt to recognizing new patterns when combined with the HOG and LBP features. After visualizing the locations and feature responses of the SVM detector, we observed that image intensity curvatures are crucial for differentiating similar distributions of vertical edges found in images of buildings and humans, and they are also useful for distinguishing pedestrian and non-pedestrian images that have the same LBP patterns. We discovered that each dimension of the Hessian feature is tuned to detecting different important patterns that previous features miss out. We had shown that when combined with existing HOG-LBP features, it could achieve over 10% improvement for most datasets.

In chapter 4, we presented our work on pedestrian density distribution modeling in a mall. Our model attempts to model the probabilities of finding a pedestrian in different nodes on a floor by modeling a series of conditional probabilities that share a Markovian relationship. These

conditional probabilities include the probabilities of pedestrians choosing different paths based on learned preference parameters associated with various path descriptors, the probabilities of pedestrians entering and leaving different portals, as well as the probabilities that relate to the latent popularities of various shops and shop categories. The model parameters are learned directly from ground truth pedestrian counts and do not require any form of tracking or trajectory data. This model is justified by its ability to consistently outperform the best uninformed prior of a uniform multinomial distribution. In fact, it turned out that we can generalize our model parameters learned from one mall to another mall with vastly different layouts and still achieved better pedestrian density distribution prediction results than the baseline. In our experiments, we discovered that regularization of path preference parameters with priors from norms in architecture theory leads to better test results as our model has many parameters to be learned and will otherwise overfit the limited observations that we have collected. The path preference parameters learned during training indicate a preference for paths with greater line of sight and smaller turn distances, which fits our intuitive understanding of human psychology and norms from architecture theory. Surprisingly, there is no preference for paths with greater expanse and shorter path distances, which we suspect is due to the correlation of expanse to line of sight and path distance to turn distance, and only the first two path descriptors are needed to model path preferences. We further experimented with refining the popularities of non-area vessels based on test data to investigate whether the model has sufficient power to fit the ground truth pedestrian counts. This led to substantial improvement in the results which implied that our model has sufficient power but it is unable to generalize the non-area vessel popularities from training data, and requires the architect to interactively set the popularities of non-area vessels. Additionally, our model is much faster and more accurate than agent-based pedestrian simulation.

In chapter 5, we extended the model for vessel popularities such that while their corresponding flow densities remain invariant within categories, they are allowed to be different across

categories, in keeping with our observations of ground truth customer flow densities that differ from category to category. The category level flow densities are learned as latent parameters from ground truth pedestrian count data. However when comparing the accuracy of pedestrian count predictions, the added complexity did not improve on the previous model which assumes constant flow density throughout. In a further analysis, we compared the latent popularities estimated by various models to ground truth area vessel category flows collected from two floors. The AreaPop model turned out to be the most accurate despite the incorrect assumption of constant flow density. Although the IndepQ and CommonQ models are less accurate than the AreaPop model, nevertheless their popularities correlate with the ground truth flow densities which suggest that there is potential to learn flow densities from ground truth pedestrian counts. The theoretically more appropriate CommonQ model underperformed most likely due to the difficulty of estimating a large number of parameters with limited data, and this is borne out by the greater accuracy of the IndepQ model which is trained directly on test data. A potential solution to these problems is an extensive data collection that may be carried out in the future.

6.2 Future Work

The potential future work discussed here is divided into three parts — data acquisition, modeling and automatic optimization of building layouts.

6.2.1 Data Acquisition

For data acquisition, we can improve pedestrian detection by encoding human visual data at a higher level of representation, in terms of region similarity instead of only local edges as in HOG, utilizing some of the research work that has been carried out on image segmentation. We can further attempt to recognize these part-based shapes in conjunction with our existing features, to provide greater accuracy in detecting pedestrians.

We can further extend the pedestrian detection framework in chapter 3 so that it can be used to support the collection of pedestrian counts needed to train the pedestrian model in chapter 4. The current process involves an experimenter capturing extended videos from a mobile camera while walking through a shopping mall, which is then followed by a lengthy and laborious process of watching the complete videos, manually counting pedestrians and annotating a map with pedestrian counts at different locations. If we are able to achieve accurate detection of pedestrians in these videos, it would substantially increase the post processing speed and reduce the effort required by the user, thus facilitating a more extensive collection of data. However pedestrian detection is currently unreliable for images taken from cameras moving in crowded, unconstrained environments due to changing backgrounds and camera directions with inter-person occlusion and motion blur. While pedestrian detection only determines the image position of a person, we also need to determine the 3D location of the person in the building to completely automate the annotation. These new challenges create future research opportunities to develop newer algorithms to tackle these problems.

6.2.2 Modeling

For modeling, we can improve our model by adding inter-floor information instead of modeling each floor separately, for example by explicitly annotating how escalators connect across floors. This is because the layouts and pedestrian densities of one floor can influence another floor. For example a popular shop near an escalator may increase the popularity of the escalator. This may make the opposite end of the escalator on the other floor become more popular than is otherwise predicted from the shop layout of the floor.

A current limitation of our model is that we assume the source and sink vessel popularities are to be mostly independent, as per (4.12). However this is unlikely to be true, e.g. a person who visits a cafe may be unlikely to subsequently visit a fast food restaurant. We also had a

simplified assumption that all portals are bidirectional, which is not entirely true as escalators are unidirectional. Future work may include modeling the source and sink vessel popularities jointly.

We can also relax our model assumptions to include variations of densities across the day (e.g. during peak and off-peak hours). These variations will not only lead to changes in total number of people but also route choice and portal popularity patterns. In addition, we may also want to consider how route choices are affected by pedestrian densities, which is not accounted for in our current model. Activities such as window shopping, loitering, queuing and sitting on benches may also influence pedestrian density distribution, which is ignored in our current model.

These extensions would require substantially more data than we have currently collected. For these to be successful, we will need to mount additional data collection efforts, which will be easier if automated pedestrian detection as discussed in section 6.2.1 is eventually implemented.

We can increase modeling scope over time and to a city-wide scale. For such a scale, it may be too expensive to collect pedestrian counts at every location in the city. One approach is to develop new techniques to learn a pedestrian density model with fewer measurement points by applying better crafted constraints on expected walking paths.

We can also look into detecting problems in the linkage of pathways and roads by collecting counts at certain areas, similar to existing techniques for detecting lossy links on the internet by collecting traffic counts at end points using network tomography [114].

6.2.3 Automatic Optimization of Building Layouts

In our current scenario, the architect is expected to create a building layout and then interactively apply the framework to visualize the pedestrian distributions while manually modifying

the layout. An alternative scenario would be to have a system that takes a draft input layout, and then optimizes the layout based on a number of desired criteria such as walkability, space volume, aesthetic, bonding and footfall. Such optimizations may be attempted through the use of multi-criteria genetic algorithms [37], with access to a library of desirable layout solutions from existing designs. This idea is somewhat controversial because it may reduce the amount of creative input by architects, but conversely it allows for a larger number of non-expert users to generate better designs without the need for access to expensive architectural and planning expertise, e.g. in economically poorer areas.

Appendix A

Appendix

A.1 Proof that Uniform Distribution is the Highest Expected Log Likelihood Value from Counts Sampled from Any Possible Multinomial Distribution

Assume the ground truth pedestrian density is uniform distributed, that means the ground truth $P(\mathbf{g})$ is a Dirichlet distribution of concentration parameter of 1:

$$P(\mathbf{g}) \sim \mathrm{Dir}(\alpha = 1). \tag{A.1}$$

The mean value from the above Dirichlet distribution is

$$\begin{aligned} \mathbf{E}[g_i] &= \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j} \\ &= \frac{1}{\sum_{j=1}^{N} 1} \\ &= 1/N \end{aligned} \tag{A.2}$$

which is a constant. N is the number of nodes in the floor layout and $\mathbf{E}[g_i]$ is the expectation of random variable g_i.

The maximum likelihood P(node) (represents as $\mathbf{p}$, p_i is element i of vector $\mathbf{p}$) values that maximize uniform ground truth density counts are

$$\max_{\mathbf{p}} T\mathbf{E}[\mathbf{g}^T \log(\mathbf{p})] \tag{A.3}$$

where T is total number of people in that floor.

Given a case of two ground truth density count $\mathbf{m}^{(1)}$ and $\mathbf{m}^{(2)}$, the maximum likelihood of both counts are:

$$\begin{aligned} &\max_{\mathbf{p}} \sum 0.5m_i^{(1)} \log(p_i) + 0.5m_i^{(2)} \log(p_i) \\ =&\max_{\mathbf{p}} \sum (0.5m_i^{(1)} + 0.5m_i^{(2)}) \log(p_i), \end{aligned} \tag{A.4}$$

which is $(0.5\mathbf{m}^{(1)} + 0.5\mathbf{m}^{(2)})/T$ where $\sum m_i^{(1)} = T$ and $\sum m_i^{(2)} = T$.

Therefore with mean of Dirichlet distribution with concentration parameter 1 is $1/N$,

$$\max_{\mathbf{p}} T\mathbf{E}[\mathbf{g}^T \log(\mathbf{p})] \tag{A.5}$$

and the solution leads to

$$\begin{aligned} p_i &= T\mathbf{E}[g_i] \\ &= \frac{T}{N} \end{aligned} \tag{A.6}$$

which is a uniform value where n is the number of nodes in that floor.

A.2 Interleaved Optimization Algorithm for Learning Path Preference Parameters $(\mathbf{a};\mathbf{b};\mathbf{c})$ and Non-Area Vessel Category Popularities f

The algorithms below are used for learning the parameters.

Algorithm to learn parameters (using interleaved training of two sets of parameters)

Input:Ground truth **m**

Output:a_i, b_i, c_i, $P(\gamma_{in}, \gamma_{out})$

set a_i, b_i, c_i and $P(\gamma_{in}, \gamma_{out})$ to their prior values

repeat

fix $P(\gamma_{in}, \gamma_{out})$ and learn a_i, b_i, c_i (see algorithm below)

fix a_i, b_i, c_i and learn $P(\gamma_{in}, \gamma_{out})$ (see algorithm below)

until Likelihood ratio with ground truth **m** and predicted $P(\zeta)$ converges (see section 4.7.3)

Algorithm to learn path descriptor parameters a_i, b_i, c_i

Input:Ground truth **m**,a_i, b_i, c_i,$P(\gamma_{in}, \gamma_{out})$

Output:a_i, b_i, c_i

repeat

find $P(\eta|\vartheta_{in}, \vartheta_{out})$ using a_i, b_i, c_i (see section 4.3.3)

find $P(\zeta) = \sum_{\substack{\eta \\ \gamma_{in},\gamma_{out} \\ \vartheta_{in},\vartheta_{out}}} P(\zeta|\eta)P(\eta|\vartheta_{in}, \vartheta_{out})P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out})P(\gamma_{in}, \gamma_{out})$

find likelihood ratio with $P(\zeta)$ and ground truth **m**(see section 4.7.3)

apply one step of gradient descent of a_i, b_i, c_i

until Likelihood ratio converges

Algorithm to learn vessel-to-vessel parameters $P(\gamma_{in}, \gamma_{out})$

Input:Ground truth **m**,a_i, b_i, c_i,$P(\gamma_{in}, \gamma_{out})$

Output:$P(\gamma_{in}, \gamma_{out})$

repeat

find $P(\gamma_{in}, \gamma_{out})$(see (4.12))

find $P(\zeta) = \sum_{\substack{\eta \\ \gamma_{in},\gamma_{out} \\ \vartheta_{in},\vartheta_{out}}} P(\zeta|\eta)P(\eta|\vartheta_{in},\vartheta_{out})P(\vartheta_{in},\vartheta_{out}|\gamma_{in},\gamma_{out})P(\gamma_{in},\gamma_{out})$

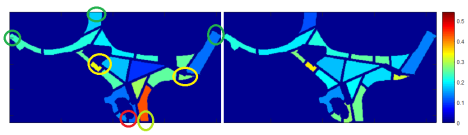
find likelihood ratio with $P(\zeta)$ and ground truth **m**(see section 4.7.3)

apply one step of gradient descent of $P(\gamma_{in})$

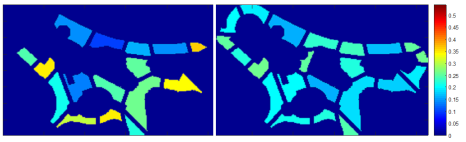
until Likelihood ratio converges

A.3 Visual Prediction Results from Learned Vessel Category Popularities f_γ

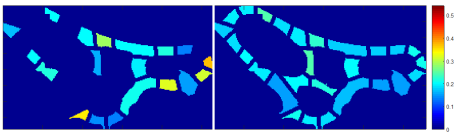
Prediction results (of section 4.8.1) can be found in the figure below:



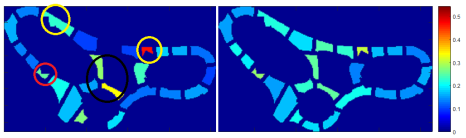
(a) Ion Floor 1



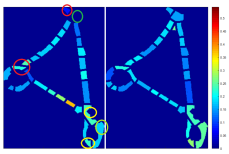
(b) Ion Floor 2



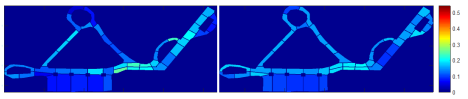
(c) Ion Floor 3



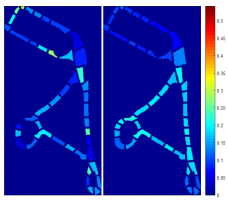
(d) Ion Floor 4



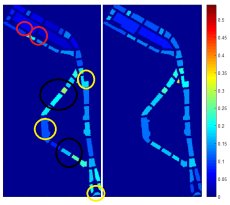
(e) Ion Floor B1



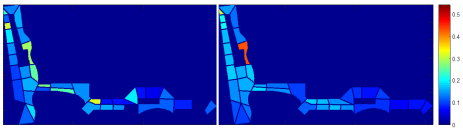
(f) Ion Floor B2



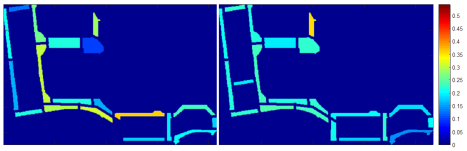
(g) Ion Floor B3



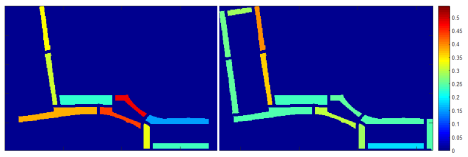
(h) Ion Floor B4



(i) Novena Square Floor 1



(j) Novena Square Floor 2



(k) Novena Square Floor 3

Figure A.1: Visual results (Left:Ground truths $P(\zeta)$ Right:Predicted pedestrian density distributions $P(\zeta)$ display on shopping mall layouts from learned **f** (see chapter 4.8.1) and path descriptor weights **a**, **b** and **c** (Need to be viewed in color, each $P(\zeta)$ is monotonically transformed to make comparison easier).

A.4 Graphical Interpretation of Neyman Pearson Test

The mathematics of Neyman Pearson test are

Maximum Likelihood of Uniform Prediction,

$$P(\mathbf{k}|\mathbf{p_N}) \propto \prod_i p_{N_i}^{k_i}$$

Maximum Likelihood of Our Model Prediction,

$$P(\mathbf{k}|\mathbf{p_A}) \propto \prod_i p_{A_i}^{k_i}$$

Likelihood Ratio,

$$R = \prod_i (\frac{p_{N_i}}{p_{A_i}})^{k_i}$$

Log Likelihood Ratio,

$$L = \sum_i k_i(\log(p_{N_i}) - \log(p_{A_i}))$$

Probability of a randomly generated sample smaller than threshold of log(0),

$$P(L < 0|\mathbf{k}, \mathbf{p_N}, \mathbf{p_A}) \begin{cases} 1 \\ 0 \end{cases}$$

depending on $\mathbf{k}, \mathbf{p_N}, \mathbf{p_A}$

p-value for threshold of log(0),

$$P(L < 0|\mathbf{p_N}, \mathbf{p_A}) = \sum_i P(L < 0|\mathbf{k_i}, \mathbf{p_N}, \mathbf{p_A}) P(\mathbf{k_i}) \tag{A.7}$$

Where:

- $\mathbf{p_N}$: Null hypothesis (Uniform Prediction), $\mathbf{p}$ is a vector and p_i is element i of the vector
- $\mathbf{p_A}$: Alternate hypothesis (Our Model Prediction)
- $\mathbf{k}$: Samples from Dirichlet-Multinominal Distribution
- $\mathbf{p_S}$: Dirichlet Distribution

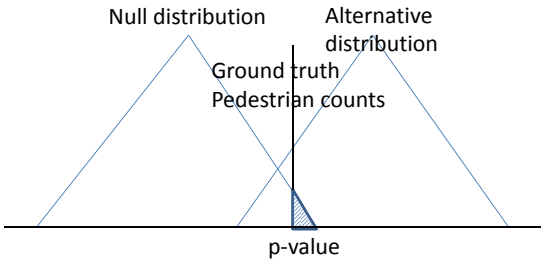


Figure A.2: Graphical interpretation of p-value of Neyman Pearson Test. n_{pass} is the shaded area, and $n_{pass} + n_{fail}$ is the total area of null distribution

First a sample is generated from Dirichlet-Multinominal Distribution. Then the sample is tested against a fixed threshold $P(L < \text{threshold}|\mathbf{k}, \mathbf{p_N}, \mathbf{p_A})$ and is 1 if it satisfies the test. Then we generate many other samples to test against the threshold. Next, find the p-value using $P(L < \text{threshold}|\mathbf{p_N}, \mathbf{p_A})$ to calculate the number of samples that satisfies the test divided by the total number of samples.

Figure A.2 shows the graphical interpretation of Neyman Pearson Test.

A.5 Theoretical Timing for Generating Steady State Pedestrian Density Distribution

The theoretical timing for generating least probable source sink in ninety percent chance:

$$(1 - P(A|B) \times P(B))^m = 0.1 \tag{A.8}$$

where $P(A|B)$ is the probability of taking the least probable path of the least probable source sink. The term $P(B)$ is the probability of choosing the least probable source sink. The term m is number of steps to generate the least probable source sink with 0.9 probability. With an agent takes on average about 6 seconds to move from one source to another sink,

$$P(A,B) = P(A|B) \times P(B) = 0.05 \times (9.8246e-04) = 4.9123e-05 \tag{A.9}$$

leads to

$$m = 4.6873e+04. \tag{A.10}$$

With 100 agents are simulated simultaneously

$$(4.6873e+04)/100 \times 6/60 = 46.873 minutes. \tag{A.11}$$

This timing is calculated based on layout shown in figure 4.18. Note that this is a difficult layout with lots of inter-connected free space at the bottom. From the above result, we should run the space syntax based pedestrian simulation for 47 minutes which is much longer than the time to generate all paths (3 minutes) in our model. Our model can be sped up dramatically by implementing using a C++ codes instead of using the original Matlab codes.

A.6 Algorithm to Calculate the Total Likelihood Ratio of q_γ Across All the Training Floors

Algorithm to calculate the total likelihood ratio of q across all the training floors

Input:$\mathbf{q}$,$P_{area}^{(floor)}$,$P_{node|in,out}$,$P_{in,out|Cin,Cout}$,$P_{GT}^{(floor)}$,$n^{(floor)}$

Output:*cost*

$cost \leftarrow 0$

for all $floor \in training\_floor$ **do**

$PCatInOut \leftarrow$ FIND_VESSEL_CATEGORY_FLOW$(\mathbf{q}, P_{area}^{(floor)})$

$PCatInOut \leftarrow$ REDISTRIBUTE_SINGLE_VESSEL_CATEGORY_FLOW$(PCatInOut)$

$P_{pred}^{(floor)} \leftarrow$ FIND_NODE_DENSITY$(P_{node|in,out}, P_{in,out|Cin,Cout}, PCatInOut)$

$cost \leftarrow cost +$ FIND_LOG_LIKELIHOOD$(P_{pred}^{(floor)}, P_{GT}^{(floor)}, n^{(floor)})$

end for

return *cost*

function $find\_vessel\_category\_flow(\mathbf{q}, P_{area}^{(floor)})$ ▷ Find P(Cin,Cout) given area and **q**

for all $Cin \in vessel\_category$ **do**

$PCatIn[Cin] \leftarrow q[Cin] \times P_{area}^{(floor)}[Cin]$

end for

$sum \leftarrow \sum_{Cin} PCatIn[Cin]$

for all $Cin \in vessel\_category, Cout \in vessel\_category$ **do**

$PCatInOut[Cin][Cout] \leftarrow PCatIn[Cin]/sum \times PCatIn[Cout]/sum$

end for

return *PCatInOut*

end function

function *redistribute_single_vessel_category_flow*(*PCatInOut*) ▷ Redistribute flow of single vessel category in P(Cin,Cout) to other categories

for all $Cin \in vessel\_category, Cout \in vessel\_category$ **do**

$PCatInOut2[Cin][Cout] = PCatInOut[Cin][Cout]$

end for

for all $Cin \in zero_P^{(floor)}, Cout = Cin$ **do**

$PCatInOut2[Cin][Cout] \leftarrow 0$

end for

for all $Cin \in zero_P^{(floor)}, Cout \neq Cin$ **do**

$$PCatInOut3[Cin][Cout] = 0.5 \times PCatInOut[Cin][Cin] \times \frac{PCatInOut[Cin][Cout]}{\sum_{k \notin zero_P^{(floor)}} PCatInOut[Cin][k]}$$

end for

for all $Cout \in zero_P^{(floor)}, Cin \neq Cout$ **do**

$$PCatInOut4[Cin][Cout] = 0.5 \times PCatInOut[Cout][Cout] \times \frac{PCatInOut[Cin][Cout]}{\sum_{k \notin zero_P^{(floor)}} PCatInOut[k][Cout]}$$

end for

for all $Cin \in vessel\_category, Cout \in vessel\_category$ **do**

$$PCatInOut[Cin][Cout] \leftarrow PCatInOut2[Cin][Cout] + PCatInOut3[Cin][Cout] + PCatInOut4[Cin][Cout]$$

end for

return *PCatInOut*

end function

function $find\_node\_density(P_{node|in,out}, P_{in,out|Cin,Cout}, PCatInOut)$ ▷ Find P(Node)

$\mathbf{P}^{(\mathbf{floor})}_{\mathbf{pred}} \leftarrow \mathbf{P}_{\mathbf{node|in,out}} \times \mathbf{P}_{\mathbf{in,out|Cin,Cout}} \times vec(\mathbf{PCatInOut})$ ▷ matrix multiplication

return $P^{(floor)}_{pred}$

end function

function $find\_log\_likelihood(P^{(floor)}_{pred}, P^{(floor)}_{GT}, n^{(floor)})$ ▷ Find log maximum likelihood ratio

$cost \leftarrow \sum_{node} \frac{1}{n^{(floor)}} \log(P^{(floor)}_{pred}[node]) - \sum_{node} P^{(floor)}_{GT}[node] \log(P^{(floor)}_{pred}[node])$

return *cost*

end function

Description of the terms:

- $training\_floor$: Set of floors in training set
- $\mathbf{q}$: Flows per area (flow densities) of category, array of q_γ
- $P^{(floor)}_{area}[Cin]$: Normalized area of specific category $\gamma = Cin$ for a particular floor
- $PCatInOut[Cin][Cout]$: Vessel category popularities $P(\gamma_{in}, \gamma_{out})$
- $\mathbf{P}_{\mathbf{node|in,out}}, \mathbf{P}_{\mathbf{in,out|Cin,Cout}}$: $P(\zeta|\vartheta_{in}, \vartheta_{out})$ and $P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out})$ respectively
- $zero^{(floor)}_P$: Set of category id of a particular floor for which there is only one vessel in the category
- $P^{(floor)}_{pred}[node], P^{(floor)}_{GT}[node]$: Predicted $P(\zeta)$ and ground truth $P(\zeta)$
- $n^{(floor)}$: Number of nodes in a particular floor

- $vec(\mathbf{M})$: Convert matrix $\mathbf{M}$ to vector (note that matrix and vector are represented as bold letter)

The algorithm shown above describes how to compute the log maximum likelihood ratio of $P(\zeta)$ for learning the parameter $\mathbf{q}$. There are four parts to this algorithm. The first part (find_vessel_category_flow function) is to compute the $P(\gamma_{in})$ by multiplying flow densities with areas of categories and doing outer product to get $P(\gamma_{in}, \gamma_{out})$. The second part (redistribute_single_vessel_category_flow function) is to redistribute flows from categories with only one vessel to all other vessels in other categories. This is to solve the technical problem when there is only one vessel in a particular category and based on the logic of $P(\vartheta_{in}, \vartheta_{out}|\gamma_{in}, \gamma_{out})$, pedestrians are not allowed to move from the same vessel to the same vessel. This makes common sense as pedestrians will unlikely to move to and from the same vessel. However some missing inter-vessel flow occurs if we do not re-distribute the flow of the single vessel to other vessels. The third part (find_node_density and find_log_likelihood functions) is to compute the predicted $P(\zeta)$ from $P(\gamma_{in}, \gamma_{out})$ and find the log maximum likelihood ratio of that predicted $P(\zeta)$ and add to the total cost. Lastly is to return the total cost (as *cost*) for the gradient descent algorithm to learn parameter $\mathbf{q}$.

For inference, we used the same algorithm above except that we returned $P(\zeta)$ of each floor instead of cost.

References

[1] BMW Automobiles. www.bmw.com. Accessed: 2016-06-02.

[2] Ion Shopping Mall. http://www.ionorchard.com/. Accessed: 2016-06-02.

[3] Mobileye. www.mobileye.com. Accessed: 2016-06-02.

[4] Novena Square Mall. http://www.velocitynovena.com/. Accessed: 2016-06-02.

[5] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, 2011.

[6] Ejaz Ahmed, Michael Jones, and Tim K. Marks. An improved deep learning architecture for person re-identification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[7] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 2211–2218, Washington, DC, USA, 2014. IEEE Computer Society.

[8] Saad Ali and Mubarak Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[9] Saad Ali and Mubarak Shah. Floor fields for tracking in high density crowd scenes. In *Proceedings of IEEE European Conference on Computer Vision*, ECCV, pages 1–14, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] Majid Alivand, Hartwig Hochmair, and Sivaramakrishnan Srinivasan. Analyzing how travelers choose scenic routes using route choice models. *Computers, Environment and Urban Systems*, 50:41 – 52, 2015.

[11] James F. Allen. Readings in qualitative reasoning about physical systems. chapter Maintaining knowledge about temporal intervals, pages 361–372. ACM, San Francisco, CA, USA, 1990.

[12] James F. Allen and George Ferguson. Actions and events in interval temporal logic. Technical report, Rochester, NY, USA, 1994.

[13] Yasuo Asakura, Eiji Hato, and Masuo Kashiwadani. Origin-destination matrices estimation model using automatic vehicle identification data and its application to the han-shin expressway network. *Transportation*, 27(4):419–438, 2000.

[14] Gideon D.P.A. Aschwanden, Simon Haegler, Frdric Bosch, Luc Van Gool, and Gerhard Schmitt. Empiric design evaluation in urban planning. *Automation in Construction*, 20(3):299 – 310, 2011.

[15] Gideon D.P.A. Aschwanden, Tobias Wullschleger, Hanspeter Mller, and Gerhard Schmitt. Agent based evaluation of dynamic city models: A combination of human decision processes and an emission model for transportation based on acceleration and instantaneous speed. *Automation in Construction*, 22:81 – 89, 2012. Planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[16] Egil Bae, Juan Shi, and Xue-Cheng Tai. Graph cuts for curvature based image denoising. *IEEE Transactions on Image Processing*, 20(5):1199–1210, 2011.

[17] Soumya Banarjee, Crina Grosan, and Ajith Abraham. Emotional ant based modeling of crowd dynamics. In *Proceedings of the 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC '05, pages 279–, Washington, DC, USA, 2005. IEEE Computer Society.

[18] Aharon Bar-Hillel, Dan Levi, Eyal Krupka, and Chen Goldberg. Part-based feature synthesis for human detection. In *Proceedings of IEEE European Conference on Computer Vision*, Crete, Greece, 2010.

[19] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc J. Van Gool. Fast stixel computation for fast pedestrian detection. In *Proceedings of IEEE European Conference on Computer Vision Workshops*, pages 11–20, Firenze,Italy, 2012.

[20] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc J. Van Gool. Pedestrian detection at 100 frames per second. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2903–2910, Providence, RI, 2012. IEEE.

[21] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? In *CVRSUAD, European Conference on Computer Vision workshop*, 2014.

[22] Aaron F. Bobick and Andrew D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, 1997.

[23] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[24] B. A. Boghossian and S. A. Velastin. Motion-based machine vision techniques for the management of large crowds. In *Proceedings of IEEE International Conference on Electronics, Circuits and Systems*, volume 2, pages 961–964 vol.2, Pafos, Cyprus, August 2002.

[25] David C. Brogan and Nicholas L. Johnson. Realistic human walking paths. In *16th International Conference on Computer Animation and Social Agents, CASA 2003, New Brunswick, NJ, USA, May 7-9, 2003*, page 94, 2003.

[26] Gabriel J. Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 2006.

[27] Drazen Brscic, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita. Person tracking in large public spaces using 3-d range sensors. *IEEE Transactions on Human-Machine Systems*, 43(6):522 – 534, 2013.

[28] Antoni B. Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, Anchorage, AK, 2008.

[29] Antoni B. Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *Proceedings of IEEE International Conference on Computer Vision*, pages 545–551, Kyoto, 2009. IEEE.

[30] Anthony Chen, Chao Yang, Sirisak Kongsomsaksakul, and Ming Lee. Network-based accessibility measures for vulnerability analysis of degradable transportation networks. *Networks and Spatial Economics*, 7(3):241–256, 2007.

[31] Anil Cheriyadat and Richard J. Radke. Detecting dominant motions in dense crowds. *Journal Selected Topics in Signal Processing*, 2(4):568–581, 2008.

[32] Arthur Daniel Costea and Sergiu Nedevschi. Word channel based multiscale pedestrian detection without image resizing and using only one classifier. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 866–893, San Diego, CA, USA, 2005.

[34] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of IEEE European Conference on Computer Vision*, Graz, Austria, 2006.

[35] T. Darrell and A. Pentland. Space-time gestures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, 1993.

[36] Anthony C. Davies, Jia Hong Yin, and Sergio A. Velastin. Crowd monitoring using image processing. *Electronics and Communication Engineering Journal*, 7:37–47, 1995.

[37] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, PPSN VI, pages 849–858, London, UK, UK, 2000. Springer-Verlag.

[38] Piotr Dollar, Serge Belongie, and Pietro Perona. The fastest pedestrian detector in the west. In *British Machine Vision Conference*, Aberystwyth, UK, 2010.

[39] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, Beijing, China, 2005.

[40] Piotr Dollar, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *Proceedings of the British Machine Vision Conference*, pages 91.1–91.11. BMVA Press, 2009. doi:10.5244/C.23.91.

[41] Piotr Dollar, Zhuowen Tu, Hai Tao, and Serge Belongie. Feature mining for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[42] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[43] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[44] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Proceedings of IEEE International Conference on Computer Vision*, Rio de Janeiro, 2007.

[45] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. Robust multi-person tracking from a mobile platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1831–1846, 2009.

[46] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[47] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2009.

[48] Pete Ferguson, Eva Friedrich, and Kayvan Karimi. Origin-destination weighting in agent modelling for pedestrian movement forecasting. In *Proceedings of the international space syntax symposium*, 2012.

[49] R.A. Fisher. *Statistical Methods For Research Workers*. Cosmo study guides. Cosmo Publications, 1925.

[50] Lawrence D. Frank, James F. Sallis, Terry L. Conway, James E. Chapman, Brian E. Saelens, and William Bachman. Many Pathways from Land Use to Health: Associations between Neighborhood Walkability and Active Transportation, Body Mass Index, and Air Quality. *Journal of the American Planning Association*, 72(1):75–87, March 2006.

[51] D. M. Gavrila and L. S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *IEEE Computer Society International Workshop on Automatic Face and Gesture Recognition*, pages 272–277, Zurich, Switzerland, 1995.

[52] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, 2007.

[53] D.M. Gavrila. Pedestrian detection from a moving vehicle. In *Proceedings of IEEE European Conference on Computer Vision*, Dublin, Ireland, 2000.

[54] Weina Ge and Robert T. Collins. Marked point processes for crowd counting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009.

[55] Weina Ge and Robert T. Collins. Crowd detection with a multiview sampler. In *Proceedings of IEEE European Conference on Computer Vision*, ECCV, pages 324–337, Berlin, Heidelberg, 2010.

[56] Nagia Ghanem, Daniel DeMenthon, David Doermann, and Larry Davis. Representation and recognition of events in surveillance video using petri nets. In *Proceedings of the*

2004 Conference on Computer Vision and Pattern Recognition Workshop, volume 7 of *CVPRW '04*, pages 112–, Washington, DC, USA, 2004. IEEE Computer Society.

[57] Percival Goodman. Architecture responsive to human needs and the ecological imperative. *JAE*, 35(1):46–50, 1981.

[58] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1395–1402, Beijing, 2005.

[59] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2012–2019, Miami, Florida, 2009.

[60] Stephen J. Guy, Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming C. Lin, and Dinesh Manocha. Pledestrians: A least-effort approach to crowd simulation. In *Proceedings of the 2010 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2010, Madrid, Spain, 2010*, pages 119–128, 2010.

[61] Flurin S Hänseler, Nicholas A Molyneaux, and Michel Bierlaire. Estimation of pedestrian origin-destination demand in train stations. Technical report, 2015.

[62] Dirk Helbing. Models for pedestrian behavior. *eprint arXiv:cond-mat/9805089*, May 1998.

[63] Dirk Helbing and Anders Johansson. *Pedestrian, crowd, and evacuation dynamics*, volume 16, pages 6476–6495. Springer, New York, 2009.

[64] Dirk Helbing and Anders Johansson. Pedestrian, crowd and evacuation dynamics. In *Encyclopedia of Complexity and Systems Science*, pages 6476–6495. Springer, 2009.

[65] Dirk Helbing, Anders Johansson, and Habib Zein Al-Abideen. Dynamics of crowd disasters: An empirical study. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 75(4):046109, 2007.

[66] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, pages 4282–4286, 1995.

[67] Bill Hillier. *Space is the Machine: A Configurational Theory of Architecture*. Cambridge University Press, 1998.

[68] Bill Hillier. *The City as a Socio-technical System: A Spatial Reformulation in the Light of the Levels Problem and the Parallel Problem*, pages 24–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[69] Roger L. Hughes. The flow of human crowds. *Annual Review of Fluid Mechanics*, 35(1):169–182, 2003.

[70] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[71] Julio Cezar Silveira Jacques Junior, Soraia Raupp Musse, and Claudio Rosito Jung. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27:66–77, 2010.

[72] Julio Cezar Silveira Jacques Jr., Adriana Braun, John Soldera, Soraia Raupp Musse, and Claudio Rosito Jung. Understanding people motion in video sequences using voronoi diagrams: Detecting and classifying groups. *Pattern Analysis Applications*, 10(4):321–332, October 2007.

[73] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal shape and flow correlation for action recognition. In *In 7th International Workshop on Visual Surveillance*, Minneapolis, MN, 2007.

[74] Saad M. Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *Proceedings of IEEE European Conference on Computer Vision*, Graz, Austria, 2006.

[75] Ansgar Kirchner and Andreas Schadschneider. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A Statistical Mechanics and its Applications*, 312:260–276, September 2002.

[76] D. Kong, D. Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *Proceedings of IEEE 18th International Conference on Pattern Recognition*, volume 3, pages 1187–1190, Hong Kong, 2006.

[77] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, Miami, FL, 2009. IEEE.

[78] Barbara Krausz and Christian Bauckhage. Loveparade 2010: Automatic video analysis of a crowd disaster. *Computer Vision Image Understanding*, 116(3):307–319, March 2012.

[79] T. Kretz and M. Schreckenberg. F.a.s.t. floor field and agent based simulation tool, 2006.

[80] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *Communication Magazine*, 48(9):140–150, September 2010.

[81] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *Proceedings of IEEE International Conference on Computer Vision*, pages 432–439, Nice, France, 2003.

[82] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, Benjamin Rozenfeld, Inria Rennes, Irisa Inria Grenoble, and Lear Ljk. Learning realistic human actions from movies. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[83] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Who are you with and where are you going? In *International Conference on Computer Vision Workshops*, pages 120 – 127, Barcelona, 2011.

[84] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 18(10):959–971, October 1996.

[85] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1,3):259–289, 2008.

[86] Bastian Leibe, Konrad Schindler, and Luc J. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, 2007.

[87] Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.

[88] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *International Journal Robotics Research*, 26(1):119–134, January 2007.

[89] Zhe Lin and Larry S. Davis. A pose-invariant descriptor for human detection and segmentation. In *Proceedings of IEEE European Conference on Computer Vision*, Marseille, France, 2008.

[90] Ping Luo, Yonglong Tian, Xiaogang Wang, and Xiaoou Tang. Switchable deep network for pedestrian detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[91] Ruihua Ma, Liyuan Li, Weimin Huang, and Qi Tian. On pixel count based crowd density estimation for visual surveillance. In *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems, 2004.*, pages 170–173, Singapore, 2005. IEEE.

[92] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel svms is efficient. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[93] A. Marana, L. da Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision*, pages 354+, Rio de Janeiro, 1998. IEEE Computer Society.

[94] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942, Miami, FL, 2009. IEEE.

[95] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal Computer Vision*, 65(1-2):43–72, 2005.

[96] D. Minnen, I. Essa, and T. Starner. Expectation grammars: leveraging high-level expectations for activity recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[97] Yoichi Miyawaki, Hajime Uchida, Okito Yamashita, Masa aki Sato, Yusuke Morito, Hiroki C. Tanabe, Norihiro Sadato, and Yukiyasu Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60(5):915 – 929, 2008.

[98] Darnell Moore and Irfan Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proceedings of 18th National Conference on Artificial intelligence*, pages 770–776, Edmonton, Alberta, Canada, 2002.

[99] J.A. Morente-Molinera, I.J. Prez, M.R. Urea, and E. Herrera-Viedma. On multi-granular fuzzy linguistic modeling in group decision making problems: A systematic review and future trends. *Knowledge-Based Systems*, 74:49 – 60, 2015.

[100] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 28(11):1863–1868, 2006.

[101] S. R. Musse and D. Thalmann. A model of human crowd behavior: Group interrelationship and collision detection analysis. In *Proceeding Workshop of Computer Animation and Simulation of Eurographics97*, pages 39–51, Budapest, Hungary, 1997.

[102] Yanghee Nam, Nwangyun Wohn, and Hyung Lee-Kwang. Modeling and recognition of hand gesture using colored petri nets. *IEEE Transactions on System Man Cybernetics Part A*, 29(5):514–521, September 1999.

[103] Hu Nan. *Spatial Temporal Patterns and Pedestrian Simulation*. PhD thesis, Nanyang Technological University, Singapore, 2014.

[104] J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

[105] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[106] Nam T. Nguyen, Dinh Q. Phung, Svetha Venkatesh, and Hung Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 955–960, San Diego, CA, USA, 2005.

[107] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-fei. Unsupervised learning of human action categories using spatial-temporal words. In *British Machine Vision Conference*, Edinburgh, UK, 2006.

[108] Ryuzo Ohno and Yohei Wada. Testing guide signs visibility for pedestrians in motion by an immersive visual simulation system. In Stefan Müller Arisona, Gideon Aschwanden, Jan Halatsch, and Peter Wonka, editors, *Digital Urban Modeling and Simulation*, volume 242 of *Communications in Computer and Information Science*, pages 339–346. Springer Berlin Heidelberg, 2012.

[109] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J. Little, and David G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of IEEE European Conference on Computer Vision*, pages 28–39, Prague, Czech Republic, 2004.

[110] Nuria M. Oliver, Barbara Rosario, and Alex P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[111] Sean O'Sullivan and John Morrall. Walking distances to and from light-rail transit stations. *Transportation Research Record: Journal of the Transportation Research Board*, 1538:19–26, 1996.

[112] Patrick Ott and Mark Everingham. Implicit color segmentation features for pedestrian and object detection. In *Proceedings of IEEE International Conference on Computer Vision*, Kyoto, 2009.

[113] C. Ottonello, M. Peri, C. Regazzoni, and A. Tesei. Integration of multisensor data for overcrowding estimation. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 1992.*, Chicago, IL, 1992.

[114] Venkata N. Padmanabhan, Lili Qiu, and Helen J. Wang. Passive network tomography using bayesian inference. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 93–94, New York, NY, USA, 2002. ACM.

[115] Xiaoshan Pan, Charles S. Han, Ken Dauber, and Kincho H. Law. Human and social behavior in computational modeling and analysis of egress. Automation in Construction, 2006.

[116] Constantine Papageorgiou and Tomaso Poggio. Trainable pedestrian detection. In *Proceedings of IEEE International Conference of Image Processing*, Kobe, 1999.

[117] Sangho Park. A hierarchical bayesian network for event recognition of human actions and interactions. *ACM Multimedia Systems Journal*, 10(2):164–179, 2004.

[118] Dawn C. Parker, Steven M. Manson, Marco A. Janssen, Matthew J. Hoffmann, and Peter Deadman. Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review. *Annals of the Association of American Geographers*, 93(2):314–337, June 2003.

[119] Gabriella Pasi and Ronald R. Yager. Modelling the concept of majority opinion in group decision making. *Information Sciences*, 176(4):390 – 414, 2006. Recent advancements of fuzzy sets: theory and practice.

[120] Nuria Pelechano, Kevin O'Brien, Barry Silverman, and Norman Badler. Crowd simulation incorporating agent psychological models, roles and communication. Technical report, DTIC Document, 2005.

[121] Nuria Pelechano, Catherine Stocker, Jan Allbeck, and Norman Badler. Being a part of the crowd: Towards validating vr crowds using presence. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 136–142, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[122] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of IEEE International Conference on Computer Vision*, pages 261–268, Kyoto, 2009.

[123] C. S. Pinhanez and A. F. Bobick. Human action detection using pnf propagation of temporal constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 898–, Santa Barbara, CA, 1998.

[124] Carlo Giacomo Prato. Route choice modeling: past, present and future research directions. *Journal of Choice Modelling*, 2(1):65–100, 2009.

[125] C. S. Regazzoni, A. Tesei, and V. Murino. A real-time vision system for crowding monitoring. In *Proceeding of International Conference on Industrial Electronics, Control, and Instrumentation*, Maui, HI, November 1993.

[126] Jens Rittscher, Peter H. Tu, and Nils Krahnstoever. Simultaneous estimation of segmentation and shape. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.

[127] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *Proceedings of IEEE International Conference on Computer Vision*, Barcelona, 2011.

[128] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Crowd counting using group tracking and local features. In *Advanced Video and Signal Based Surveillance*, Boston, MA, 2010.

[129] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1709–1718, New York, NY, USA, 2006.

[130] Michael S. Ryoo and Jake K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proceedings of IEEE International Conference on Computer Vision*, Kyoto, 2009.

[131] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[132] Andrea Saltelli. Sensitivity analysis for importance assessment. In *Proceedings of the 3rd International Symposium on Sensitivity Analysis of Model Output*, pages 3–18, 2001.

[133] Silvio Savarese, Andrey Delpozo, Juan Carlos Niebles, and Li Fei-fei. Spatial-temporal correlations for unsupervised action classification. IEEE Workshop on Motion and video Computing, 2008.

[134] Andreas Schadschneider, Ansgar Kirchner, and Katsuhiro Nishinari. CA approach to collective phenomena in pedestrian dynamics. In *Cellular Automata, 5th International Conference on Cellular Automata for Research and Industry, ACRI 2002, Geneva, Switzerland, October 9-11, 2002, Proceedings*, 2002.

[135] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. Deeply learned attributes for crowded scene understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[136] Jing Shao, Chen Change Loy, and Xiaogang Wang. Scene-independent group profiling in crowd. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[137] Eli Shechtman and Michal Irani. Space-time behavior based correlation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 405–412, San Diego, CA, USA, June 2005.

[138] Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.

[139] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Tracking multiple people under global appearance constraints. In *Proceedings of IEEE International Conference on Computer Vision*, pages 137–144, Barcelona, 2011.

[140] Jeffrey Mark Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal Artifical Intelligence Research*, 15(1):31–90, February 1999.

[141] Herb Sorensen. *Inside the Mind of the Shopper: The Science of Retailing*. Pearson Education, 2009.

[142] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, pages 265–, Coral Gables, FL, 1995.

[143] Martin Stubenschrott, Christian Kogler, Thomas Matyus, and Stefan Seer. A dynamic pedestrian route choice model validated in a high density subway station. *Transportation Research Procedia*, 2:376 – 384, 2014. The Conference on Pedestrian and Evacuation Dynamics 2014 (PED 2014), 22-24 October 2014, Delft, The Netherlands.

[144] Daisuke Sugimura, Kris M. Kitani, Takahiro Okabe, Yoichi Sato, and Akihiro Sugimoto. Using individuality to track individuals: clustering individual trajectories in crowds using local appearance and frequency trait. In *Proceedings of IEEE International Conference on Computer Vision*, Kyoto, 2009.

[145] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

[146] Alasdair Turner and Alan Penn. Making isovists syntactic: Isovist integration analysis. In *Proceedings of of Second International Symposium on Space Syntax, 29th March - 2rd*, 1999.

[147] Oncel Tuzel, Fatih Porikli, and Peter Meer. Human detection via classification on riemannian manifolds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007.

[148] Laura Vaughan. *Space Syntax Observation Manual*. University College London, UK, 2001.

[149] Ashok Veeraraghavan, Rama Chellappa, and Amit K. Roy-Chowdhury. The function space of an activity. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, USA, 2006.

[150] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proceedings of IEEE International Conference on Computer Vision*, Nice, France, 2003.

[151] Alan M. Voorhees. A general theory of traffic movement. *Transportation*, 40(6):1105–1116, 2013.

[152] Van-Thinh Vu, Francois Bremond, and Monique Thonnat. Automatic video interpretation: a novel algorithm for temporal scenario recognition. In *Proceedings of 18th international joint conference on Artificial intelligence*, IJCAI'03, pages 1295–1300, Acapulco, Mexico, 2003.

[153] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.

[154] Liming Wang, Jianbo Shi, Gang Song, and I-Fan Shen. Object detection combining recognition and segmentation. In *Proceedings of Asian Conference on Computer Vision*, 2007.

[155] Xiaogang Wang, Xiaoxu Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539–555, March 2009.

[156] Xiaoyu Wang, Tony Xu Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Proceedings of IEEE International Conference on Computer Vision*, Kyoto, 2009.

[157] Tomoki Watanabe and Satoshi Ito. Two co-occurrence histogram features using gradient orientations and local binary patterns for pedestrian detection. In *Proceedings of IEEE Asian Conference on Pattern Recognition*, 2013.

[158] Christian Wojek and Bernt Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM Symposium Pattern Recognition*, Munich, Germany, 2008.

[159] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[160] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 31(12):2115–2128, December 2009.

[161] Jianxin Wu, Christopher Geyer, and James M. Rehg. Real-time human detection using contour cues. In *Proceedings of IEEE International Conference of Robotics and Automation*, Shanghai, 2011.

[162] Xinyu Wu, Guoyuan Liang, Ka Keung Lee, and Yangsheng Xu. Crowd density estimation using texture analysis and learning. In *International Conference on Robotics and Biomimetics*, pages 214–219, Kunming, 2006.

[163] Song Xu and Henry Been-Lirn Duh. A simulation of bonding effects and their impacts on pedestrian dynamics. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):153–161, 2010.

[164] Yaser Yacoob and Michael J. Black. Parameterized modeling and recognition of activities. In *Proceedings of IEEE International Conference on Computer Vision*, Bombay, 1998.

[165] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and where are you going? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1345–1352, Providence, RI, 2011.

[166] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Champaign, IL, 1992.

[167] D.B. Yang, H.H. Gonzalez-Banos, and L.J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings of 9th IEEE International Conference on Computer Vision*, Nice, France, 2003.

[168] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Understanding pedestrian behaviors from stationary crowd groups. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[169] Alper Yilmaz and Mubarak Shah. Actions sketch: A novel action representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.

[170] Yuanhao Yu, Zhen Lei, Dong Yi, and Stan Z. Li. Detecting individual in crowd with moving feature's structure consistency. In *Proceedings of IEEE International Conference on Computer Vision Workshops*, pages 934–941, Barcelona, 2011. IEEE.

[171] A. K. Zaidi. On temporal logic programming using petri nets. *IEEE Transactions on Systems Man Cybernetics Part A*, 29(3):245–254, May 1999.

[172] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Proceedings of IEEE European Conference on Computer Vision*, volume 7573 of *Lecture Notes in Computer Science*, pages 343–356, Firenze,Italy, 2012. Springer.

[173] Lihi Zelnik-Manor and Michal Irani. Event-based analysis of video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, USA, 2001.

[174] Beibei Zhan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Journal Machine Vision and Applications*, 19(5-6):345–357, 2008.

[175] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. Modeling individual and group actions in meetings with layered hmms. *IEEE Transactions on Multimedia*, 8(3):509–520, September 2006.

[176] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[177] Shanshan Zhang, Christian Bauckhage, and Armin B. Cremers. Informed haar-like features improve pedestrian detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[178] Tao Zhao and Ram Nevatia. Bayesian human segmentation in crowded situations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[179] Suiping Zhou, Dan Chen, Wentong Cai, Linbo Luo, Malcolm Yoke Hean Low, Feng Tian, Victor Su-Han Tay, Darren Wee Sze Ong, and Benjamin D. Hamilton. Crowd modeling and simulation technologies. *ACM Transactions on Modelling and Computer Simulation*, 20(4):20:1–20:35, November 2010.

Publications

Published:

- **Sing Kuang Tan**, Tat-Jen Cham, Jianxin Wu, "Steerable second order intensity features for pedestrian detection." *Proceedings of 3rd Asian Conference on Pattern Recognition (ACPR)*, 2015.