Cooperative people detection and tracking strategies with a mobile robot and wall mounted cameras

Alhayat Ali Mekonnen

▶ **To cite this version:**

Alhayat Ali Mekonnen. Cooperative people detection and tracking strategies with a mobile robot and wall mounted cameras. Robotics [cs.RO]. Université Paul Sabatier - Toulouse III, 2014. English. NNT : . tel-01068355

HAL Id: tel-01068355

https://theses.hal.science/tel-01068355

Submitted on 25 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Systèmes embarqués et Robotique*

Présentée et soutenue par
Alhayat Ali MEKONNEN

Le *11 Mars 2014*

Titre :

Coopération de Réseaux de Caméras Ambiantes et de Vision Embarquée sur Robot Mobile pour la Surveillance de Lieux Publics

---

Ecole doctorale : *Systèmes (EDSYS)*

Unité de recherche :
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)

Directeur(s) de Thèse :
Ariane HERBULOT, Maître de Conférences, Université Toulouse III - Paul Sabatier
Frédéric LERASLE, Professeur des Universités, Université Toulouse III - Paul Sabatier

Rapporteurs :
Eric MARCHAND, Professeur des Universités, Université de Rennes 1
Thierry CHATEAU, Professeur des Universités, Université Blaise Pascal

Autre(s) membre(s) du jury :
Michel DEVY, Directeur de Recherche, LAAS-CNRS (president)
Jean-Marc ODOBEZ, Senior Research Scientist, IDIAP Research Institute

P H D T H E S I S

Prepared at

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)

Submitted for obtaining the degree of

Doctor of the University of Toulouse

Delivered by: **University of Toulouse III - Paul Sabatier**

In: **Embedded Systems and Robotics**

Ecole doctorale Systmes (EDSYS)

By: **Alhayat Ali MEKONNEN**

Cooperative People Detection and Tracking Strategies with a Mobile Robot and Wall Mounted Cameras

---

Coopération de réseaux de caméras ambiantes et de vision embarquée sur robot mobile pour la surveillance de lieux publics

---

Defended on the **11th of March 2014**

Jury Members:

Reviewers:	Eric MARCHAND	*Professor*	Université de Rennes 1
	Thierry CHATEAU	*Professor*	Université Blaise Pascal
Examiners:	Jean-Marc ODOBEZ	*Senior Res. Scientist*	IDIAP Research Institute
	Michel DEVY	*Research Director*	LAAS-CNRS
Thesis Advisors:	Ariane HERBULOT	*Maître de Conference*	Université Toulouse III
	Frédéric LERASLE	*Professor*	Université Toulouse III

Research Unit: Robotic, Action, and Perception (RAP) Team, LAAS-CNRS

Abstract

This thesis deals with detection and tracking of people in a surveilled public place. It proposes to include a mobile robot in classical surveillance systems that are based on environment fixed sensors. The mobile robot brings about two important benefits: (1) it acts as a mobile sensor with perception capabilities, and (2) it can be used as means of action for service provision. In this context, as a first contribution, it presents an optimized visual people detector based on Binary Integer Programming that explicitly takes the computational demand stipulated into consideration. A set of homogeneous and heterogeneous pool of features are investigated under this framework, thoroughly tested and compared with the state-of-the-art detectors. The experimental results clearly highlight the improvements the different detectors learned with this framework bring to the table including its effect on the robot's reactivity during on-line missions.

As a second contribution, the thesis proposes and validates a cooperative framework to fuse information from wall mounted cameras and sensors on the mobile robot to better track people in the vicinity. Finally, we demonstrate the improvements brought by the developed perceptual modalities by deploying them on our robotic platform and illustrating the robot's ability to perceive people in supposed public areas and respect their personal space during navigation.

Résumé

Actuellement, il y a une demande croissante pour le déploiement de robots mobile dans des lieux publics. Pour alimenter cette demande, plusieurs chercheurs ont déployé des systèmes robotiques de prototypes dans des lieux publics comme les hôpitaux, les supermarchés, les musées, et les environnements de bureau. Une principale préoccupation qui ne doit pas être négligé, comme des robots sortent de leur milieu industriel isolé et commencent à interagir avec les humains dans un espace de travail partagé, est une interaction sécuritaire. Pour un robot mobile à avoir un comportement interactif sécuritaire et acceptable - il a besoin de connaître la présence, la localisation et les mouvements de population à mieux comprendre et anticiper leurs intentions et leurs actions. Cette thèse vise à apporter une contribution dans ce sens en mettant l'accent sur les modalités de perception pour détecter et suivre les personnes à proximité d'un robot mobile.

Comme une première contribution, cette thèse présente un système automatisé de détection des personnes visuel optimisé qui prend explicitement la demande de calcul prévue sur le robot en considération. Différentes expériences comparatives sont menées pour mettre clairement en évidence les améliorations de ce détecteur apporte à la table, y compris ses effets sur la réactivité du robot lors de missions en ligne. Dans un deuxiè contribution, la thèse propose et valide un cadre de coopération pour fusionner des informations depuis des caméras ambiant affixé au mur et de capteurs montés sur le robot mobile afin de mieux suivre les personnes dans le voisinage. La même structure est également validée par des données de fusion à partir des différents capteurs sur le robot mobile au cours de l'absence de perception externe. Enfin, nous démontrons les améliorations apportées par les modalités perceptives développés en les déployant sur notre plate-forme robotique et illustrant la capacité du robot à percevoir les gens dans les lieux publics supposés et respecter leur espace personnel pendant la navigation.

Acknowledgment

Alhamdulilah! Praise be to the Almighty the Cherisher and Sustainer of the Worlds.

This thesis work marks an important personal accomplishment–one which I thought was far from possible! It would not have happened had it not been for all the guidance, encouragement, support, and help I had gotten from my supervisors, colleagues, staffs, friends, and family. Hence, I would like to take this opportunity to say thank you!

On a more specific note, I would like convey my warmest heartfelt thanks to: my supervisors Frédéric Lerasle and Ariane Herbulot, for all their mentorship throughout the years; the Jury members, for evaluating my thesis and for all their constructive feedback; Catherine Stasiulis, for going out of her way to ease the administrative hustle; Patrick Danès, for all his advice and hospitality; Cyril Briand, for all the fruitful collaborations; Mathieu Herrb and Anthony Mallet, for all their help on the robotic platforms; and all past and present members of the Robotic team (RAP, RIS, GEPETTO) at LAAS-CNRS. Thank you all for helping me make it through and making it such a fulfilling experience.

I also would like to say that I am infinitely and deeply grateful to my parents, Gashye (Ali) and Shenkorkit, and siblings, Ousish, Ayu, and A.B.D. This journey would not have been possible without their unconditional love and support. Last but not least, my deepest gratitude goes to my wonderful wife, Tizuye–I am wholeheartedly indebted to her for enduring all the ups and downs beside me.

Contents

List of Tables x

List of Figures xiii

List of Algorithms xvii

Abbreviations xix

1 Introduction 1
- 1.1 Background and Context . . . 1
- 1.2 Scope and Outline of the Thesis . . . 8
  - 1.2.1 Automated People Detection . . . 9
  - 1.2.2 Cooperative Multi-person Tracking . . . 10
- 1.3 Contributions and Organization . . . 10
  - 1.3.1 Contributions . . . 10
  - 1.3.2 Manuscript Organization . . . 12

I AUTOMATED PEOPLE DETECTION 13

2 Trends, Modes, and Considerations 15
- 2.1 Introduction . . . 15
- 2.2 Trends and Modes in the Literature . . . 17
  - 2.2.1 Sensors . . . 17
  - 2.2.2 Vision based Detection . . . 19
  - 2.2.3 Multi-modal Approaches . . . 28
  - 2.2.4 Discussions . . . 29
- 2.3 Visual Datasets . . . 32
  - 2.3.1 Ladybug Dataset . . . 32
  - 2.3.2 INRIA Person Dataset . . . 33
  - 2.3.3 Caltech Pedestrian Dataset . . . 34
- 2.4 Evaluation Metrics . . . 35
  - 2.4.1 PW Evaluation: Detector Error Tradeoff (DET) . . . 36

2.4.2 PW Evaluation: Precision-Recall . . . 36
2.4.3 Full Image Evaluation . . . 36
2.5 Summary . . . 37

3 Optimized HOG based Person Detector **39**
3.1 Introduction . . . 39
3.2 Framework . . . 40
3.3 HOG-based Feature Set . . . 41
3.4 Weak Learners . . . 43
3.4.1 Fisher's Linear Discriminant Analysis . . . 44
3.4.2 Support Vector Machines . . . 45
3.5 Pareto-Front Analysis . . . 45
3.6 Feature Selection via Binary Integer Programming . . . 46
3.7 Discrete AdaBoost and Cascade Construction . . . 48
3.8 Experiments and Results . . . 49
3.8.1 Implementation Details and Validation . . . 50
3.8.2 Results . . . 53
3.8.3 GPU Implementation . . . 58
3.9 Discussions . . . 58
3.10 Conclusions . . . 62

4 Mining Heterogeneous Features for Improved Detection **63**
4.1 Introduction . . . 64
4.2 Framework . . . 64
4.3 Features and Weak Classifiers . . . 65
4.3.1 Heterogeneous Feature Set . . . 65
4.3.2 Weak Classifiers . . . 68
4.3.3 Computation Time . . . 68
4.4 Nodal Strong Classifier Learning Schemes . . . 69
4.4.1 Pareto-Front and AdaBoost . . . 69
4.4.2 Binary Integer Optimization and AdaBoost . . . 70
4.4.3 AdaBoost with Random Feature Sampling . . . 70
4.4.4 Computation Time Weighted AdaBoost . . . 70
4.5 Cascade Detector Learning . . . 72
4.6 Experiments and Results . . . 72
4.6.1 Implementation Details and Validation . . . 73
4.6.2 Results . . . 74
4.7 Discussions . . . 82
4.8 Conclusions . . . 84

II COOPERATIVE PERCEPTION FOR TRACKING PEOPLE **87**

5 Cooperative Perception and Multi-person Tracking: An Overview **89**
5.1 Introduction . . . 89
5.1.1 Environment Fixed Sensors . . . 90
5.1.2 Mobile Sensors and Sensor Fusion Modes . . . 90
5.1.3 Environment Fixed and Mobile Sensors . . . 92
5.2 Multi-Person Tracking . . . 92
5.2.1 Overview . . . 92

5.2.2 Bayesian Formulation . . . 95
5.2.3 MCMC- and RJMCMC-Particle Filters . . . 96
5.2.4 Evaluation Metrics . . . 99
5.3 Conclusion . . . 100

6 Implementation of A Cooperative Perception System 101
6.1 Introduction . . . 102
6.2 Framework and Architecture . . . 102
6.2.1 Environment Configuration . . . 102
6.2.2 System Block Diagram . . . 103
6.2.3 Environment Calibration . . . 103
6.3 Perceptual Components . . . 105
6.3.1 Multi-person Detection . . . 105
6.3.2 Multi-person Tracking Implementation . . . 108
6.4 Robot Navigation Aspects . . . 113
6.4.1 Personal Space Model . . . 113
6.4.2 Nearness Diagram (ND) Navigation . . . 114
6.5 Evaluations and Results . . . 114
6.5.1 Off-line Evaluation . . . 114
6.5.2 On-line Evaluation . . . 118
6.6 Towards a Self-Contained Robotic Perceptual System . . . 124
6.7 Discussions . . . 125
6.8 Conclusion . . . 126

Conclusions and Future Prospects 131

A Brief Description of Sensors used for People Detection 135
A.1 Passive Sensors . . . 135
A.2 Active Sensors . . . 136

B Calibration Values: External cameras and a mobile robot 139

C Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU (publié dans RFIA 2014) 141
C.1 Introduction . . . 141
C.2 Descriptif de notre approche . . . 143
C.2.1 Les descripteurs . . . 144
C.2.2 Extraction du front de Pareto . . . 145
C.2.3 Sélection des descripteurs et apprentissage de la cascade . . . 145
C.3 Optimisation discrète . . . 145
C.4 Evaluations et résultats . . . 147
C.4.1 Critères d'évaluation . . . 147
C.4.2 Jeux de données . . . 147
C.4.3 Apprentissage . . . 148
C.4.4 Résultats et discussions . . . 148
C.5 Conclusions et perspectives . . . 151

D Coopération entre un robot mobile et des caméras d'ambiance pour le suivi multi-personnes (publié dans RFIA 2012) **153**
D.1 Introduction . . . 154
D.2 Description de notre architecture perceptuelle . . . 155
D.2.1 Plateforme robotique . . . 155
D.2.2 Synoptique descriptif du système . . . 155
D.3 Modalité de suivi de passants . . . 157
D.3.1 Formalisme RJMCMC – PF . . . 158
D.3.2 Implémentation . . . 158
D.4 Évaluations . . . 161
D.5 Intégration sur le robot Rackham et démonstration . . . 163
D.6 Conclusion et Perspectives . . . 164

List of Tables

2.1 Summary of different sensors with associated people detection approaches. 18

3.1 Illustration of HOG features computation. (Illustrations taken from [Dalal 2006a].) 42
3.2 Open source libraries used for implementing different components of the proposed framework. 51
3.3 Comparative summary of learned cascade classifiers on Ladybug dataset with varying FPR and Dalal and Triggs detector. 54
3.4 Comparative summary of learned cascade classifiers on INRIA dataset, using a constant per node FPR of 0.5 and TPR of 1.0, and Dalal and Triggs detector. . . 55
3.5 Computation time comparison with the state-of-the-art. 58

4.2 Summary of the cascade detector trained on the Ladybug dataset. 76
4.3 Summary of the cascade detector trained on the INRIA datasets. Miss Rate is reported at 10^{-4} FPPW. 78
4.4 Computation time comparison with the state-of-the-art. 80

6.1 Parameter values used to produce the results reported in this section. 115
6.2 Laser-based only perception. 115
6.3 Wall-mounted cameras-based only perception. 115
6.4 Cooperative perception using a single wall-mounted camera. 116
6.5 Cooperative perception using the two wall-mounted cameras. 116
6.6 Id swap occurrences in each tracking mode. 116
6.9 Cooperative perception on-line evaluation. 120
6.10 Cooperative perception on-line evaluation (MOTP and MOTA). 120
6.11 Robotic mission success. 122

C.1 Récapitulatif des descripteurs utilisés ; $\mathfrak{u} = 0.0535\mu s$. 145
C.2 Résumé du détecteur en cascade entrainé sur les bases de données de l'INRIA. Les taux de faux négatifs sont donnés à un FPPW de 10^{-4}. 149

D.1 Résultats d'évaluation du suivi (moyenne et écart-type). 162

List of Figures

1.1 Essential components of an intelligent public place surveillance system. 2
1.2 Automatic abandoned luggage detection [Porikli 2008]. 3
1.3 Robotic systems for safe inspection and removal of suspicious/abandoned luggage 3
1.4 3D floor plan of an airport terminal. 4
1.5 Examples of some perception sensors/systems. 4
1.6 Exemplary assistance robotic systems in action. 5
1.7 Mono-sensor configuration. 6
1.8 Centralized cooperation. 6
1.9 Hierarchical cooperation. 7
1.10 Decentralized cooperation. 7

2.1 Important components of a vision based person detector. 19
2.2 Illustration of an exemplar people detector learning scheme. 20
2.3 Taxonomy of visual person detection methods. 20
2.4 Implicit Shape Model. 22
2.5 Some candidate window generation variants for detecting people in images. . . . 23
2.6 Illustration of a sequential multi-modal people detector. 29
2.7 Performance of the state-of-the-art people detectors (produced using the toolbox from [Dollár 2012]) 30
2.8 *Ladybug2* camera and a corresponding stitched image. 32
2.9 Sample positive images taken from the Ladybug training dataset. 33
2.10 Sample negative images taken from the Ladybug training dataset. 33
2.11 Demonstrative positive images taken from the Ladybug test dataset. 33
2.12 Demonstrative negative images taken from the Ladybug test dataset. 33
2.13 Illustrative samples from the INRIA person dataset. 34
2.14 Illustrative image frames taken from the Caltech pedestrian dataset. 35

3.1 An attentional detector cascade configuration. 40
3.2 Feature selection and classifier learning framework used at each node of a cascade. 41
3.3 Illustration of the HOG feature pool set generation. 43
3.5 Fisher's LDA and SVM based weak learner classification illustration. 44
3.4 Two projection vectors, LDA results in the second vector, (b), as it maximizes class separation. 44

3.6 Sample extracted Pareto Front. 46
3.7 Decision Tree depth validation when used in conjunction with Fisher's LDA. . . 52
3.8 Miss rate variation as a function of the detection window overlap for non-maximal suppression. 53
3.9 DET plots showing performance on Ladybug test dataset. 54
3.10 Illustration of selected features overlaid on an average gradient image. 55
3.11 Comparative curve for selected cascade detector and Dalal and Triggs detector on the Ladybug dataset. 56
3.12 Selected features of the detectors trained on the INRIA dataset. 56
3.13 Full image evaluation results on the INRIA test dataset. For all the other approaches the results published in [Dollár 2012] are used. 57
3.14 Full image evaluation results on the Caltech test dataset. 59
3.15 Sample full image detection outputs from the Ladybug dataset. 60
3.16 Full image detection illustrations on images taken from the INRIA test dataset. . 60
3.17 Illustrative full image detections taken from the Caltech dataset. 61

4.1 Investigated cascade node training schemes using heterogeneous pool of features. 64
4.2 Feature region specification. 65
4.3 Set of extended Haar like feature types (configurations) used. 66
4.4 Illustration of a single EOH feature extraction from a given window (taken from [Gerónimo 2007]). 66
4.5 CS-LBP feature extraction steps. 67
4.6 Illustration of sample CSS features. 67
4.7 Sample Pareto-Front extraction with heterogeneous features. (Best viewed in color.) 69
4.8 Decision tree depth validation for (a) Haar like features, (b) EOH features, and (c) CS-LBP features. 73
4.9 Error rate on a validation set using the computation time weighted AdaBoost trained with different β values. 75
4.10 DET of different detectors trained and tested on the Ladybug dataset. 75
4.11 The features selected and used in the first node of the cascade with the Ladybug dataset. 76
4.12 Illustration of the different features selected on different nodes of cascade of BIP+AdaBoost trained on Ladybug dataset 77
4.13 DET of different detectors trained and tested on the INRIA dataset. 77
4.14 The features selected and used in the first node of the cascade trained on the INRIA dataset 78
4.15 Sample depictions of the heterogeneous features selected at different nodes of the cascade BIP+AdaBoost(Ad) trained on the INRIA dataset using an adaptive FPR. 79
4.16 Histogram of selected features in the first 9 nodes of the model trained on the INRIA dataset using both fixed FPR of 0.5 and adaptive FPR. 79
4.17 Comparative full image evaluation on the INRIA test set. 79
4.18 Full image evaluation results on the Caltech test dataset. 81

6.1 Cooperative perceptual platform; wall-mounted cameras (with rough positioning and fields of view) and Rackham, the mobile robot. 103
6.2 Cooperative people detection and tracking framework. 104
6.3 Cooperative system environment calibration. 105

6.4 LRF scan illustrations showing the human-robot situation in (a) and the associated laser scan in (b). Scans corresponding to legs are shown circled. Rackham is shown as the red circle in (b). . . . 106
6.5 Sample images from the two wall mounted cameras. . . . 107
6.6 HS+V histograms computed for two targets. . . . 108
6.7 Illustration of the add proposal distribution. . . . 111
6.8 Personal space models. (a) and (b) show the model based on Gaussian functions used in [Vasquez 2012], and (c) shows a simplified elliptical discrete zone model. 113
6.9 Multi-person tracking illustrations taken from sequence I. . . . 116
6.10 Multi-person tracking illustrations taken from sequence II. . . . 117
6.11 Motion capture setup. . . . 119
6.12 Multi-person tracking evaluation with variable walking speed of targets. . . . 120
6.13 Minimum distance between a mobile robot and a person's security zone as a function of the robot's speed to guarantee safe navigation. . . . 122
6.14 Experiments carried out to evaluate the robot's safe navigation that relies on the developed perceptual inputs. . . . 123
6.15 Sample robotic run for safe navigation evaluation. Perception is based on the GPU-HOG detector. (Please see text for explanation). . . . 128
6.16 Sample robotic run for safe navigation evaluation with perception based on the GPU-HOG-BIPBoost detector. (Please see text for explanation). . . . 129
6.17 A robotic self-contained perceptual system using a spherical camera, *the Ladybug2*, and laser range finder. . . . 130

A.1 Kinect sensor in use. . . . 137
A.2 Illustration of sensor data from multiple sensors mounted on a mobile robot. . . . 138

C.1 Schéma du synoptique de l'apprentissage du classifieur fort propre à chaque nœud de la cascade. . . . 142
C.2 DET des détecteurs entraînés et testés sur la base INRIA. . . . 148
C.3 Représentations d'exemples des descripteurs hétérogènes choisis. . . . 149
C.4 Histogramme de descripteurs sélectionnées pour les 9 premiers noeuds des modèles entraînés sur la base INRIA avec un FPR fixe de 0.5 et avec un FPR adaptatif. . 150
C.5 Evaluation comparative avec images complètes sur la base test de l'INRIA. . . . 150

D.1 Plateforme perceptuelle, caméras d'ambiance (positions et champs de vues associées) et robot mobile Rackham. . . . 155
D.2 Synoptique de notre architecture perceptuelle. . . . 156
D.3 Exemple de coupe laser SICK 156
D.4 Détection RFID. . . . 157
D.5 Exemple d'images acquise par les caméras déportées : champ de vue (a), segmentation des régions mobiles par $\Delta - \Sigma$ (b), détection de personnes par HOG (c). . . . 158
D.6 Exemples de suivi multi-personnes pour la séquence II. . . . 162
D.7 Zone de sécurité autour d'une cible et définie sur le plan du sol. . . . 163
D.8 Illustration du comportement de Rackham en présence de plusieurs passants. . . . 163
D.9 Illustrations issues de la poursuite d'utilisateur en évitant les passants. . . . 165

List of Algorithms

3.1 Pareto-Front Computation 46
3.2 Discrete AdaBoost Training 49
4.1 Computation Time Weighted AdaBoost 71
5.1 MCMC-PF based Multi-person Tracking (fixed M number of targets) 97
5.2 RJMCMC-PF based Multi-person Tracking (variable number of targets) 99
6.1 RJMCMC-PF based Multi-person Tracking Implementation 109

Abbreviations

2D Two Dimensional

3D Three Dimensional

ASU Average Speed Up

BIP Binary Integer Programming

CS-LBP Center-Symmetric Local Binary Pattern

CSS Color Self Similarity

CT Computation Time

DET Detector Error Tradeoff

DRLBP Discriminative Robust Local Binary Pattern

DT Decision Tree

EOH Edge Orientation Histogram

FI Full Image

FN False Negative

FOV Field of View

FP False Positive

FPPI False Positives Per Image

FPPW False Positives Per Window

FPR False Positive Rate

fps frames per second

GPU Graphical Processing Unit

HOF Histogram of Flow

HOG Histogram of Oriented Gradients

IMH Internal Motion Histograms

IR Infrared

LBP Local Binary Pattern

LDA Linear Discriminant Analysis

LRF Laser Range Finder

MBH Motion Boundary Histograms

MCL Multiple Component Learning

MR Miss Rate

MRF Markov Random Field

MS Mean Shift

ND Nearness Diagram

NMS Non-Maximal Suppression

NRLBP Non Redundant Local Binary Pattern

PM Pairwise Max

PTZ Pan-Tilt-Zoom

PW Per Window

RFID Radio Frequency Identification

RGB+D Red Green Blue + Depth

RJMCMC-PF Reversible Jump Markov Chain Monte Carlo - Particle Filter

SVM Support Vector Machine

TN True Negative

TP True Positive

TPR True Positive Rate

VOC Visual Object Classification

Chapter 1

Introduction

Contents

1.1 Background and Context 1
1.2 Scope and Outline of the Thesis 8
1.2.1 Automated People Detection 9
1.2.2 Cooperative Multi-person Tracking 10
1.3 Contributions and Organization 10
1.3.1 Contributions 10
1.3.2 Manuscript Organization 12

1.1 Background and Context

In the last two decades, many intelligent surveillance systems have proliferated and the attention given by the scientific community has increased considerably [Räty 2010]. Raty [Räty 2010] describes a surveillance system as a technological tool that assists human operators by offering an extended perception and reasoning capability about situations of interest that occur in the monitored environments. Through the years, surveillance systems have evolved from mediocre analogue systems that provide video feed to more complex systems comprised of multiple sensors and mobile robots that have the ability to automatically detect an event and provide necessary action on it. These recent advances are quite appealing as they decrease the load on human monitoring (*e.g.*, by a security personnel), which is labor intensive and inefficient, by automating the perception and event detection. As a consequence, there is an increased demand for these surveillance systems mainly in the following listed application areas:

- Public security and safety: This is the principal area attracting many researchers. The need to ensure safety in public areas like transportation hubs—*e.g.*, airport termi-

nals [Foucher 2011], railways [Ronetti 2000], maritime environments [Pozzobon 1999]—public places—*e.g.*, banks [Zambanini 2009], shopping malls [Bouma 2013], parking lots [Micheloni 2003]—by detecting anomalous activities and taking counter measures.

- Health-care: Automated patient monitoring [Rajasekaran 2010]; Monitoring activities of the elderly relieving caregivers from the need to keep vigilant eye on each cared person [Zouba 2009].
- Traffic control: Automatic traffic volume and congestion perception on motorways to assist drivers dynamically plan their trips more efficiently [Tseng 2002]; Automatic road traffic offense [Marikhu 2013] and accident [Kamijo 2000] detection for expedited response team notification.
- Assistance: To provide help for people requiring assistance automatically, for example, heavy luggage [Jayawardena 2010], direction guidance [Bennewitz 2005], etc.
- Inspection: Automated systems to inspect warehouses and storage sites, identifying anomalous situations, such as flooding and fire, detect intruders, and determine the status of inventoried objects [Everett 2003].
- Military Applications: Various military applications ranging from border surveillance, to enemy tracking, battlefield surveillance, and target classification [Arampatzis 2005].

The application domain for intelligent surveillance systems is quite vast, intended for different event identification targeting individual persons, crowds, automobile traffic, inanimate objects, *etc*. The target of a surveillance is the entity or entities upon which the surveillance operates, *i.e.*, those entities among which the event detection method aims to detect events on.

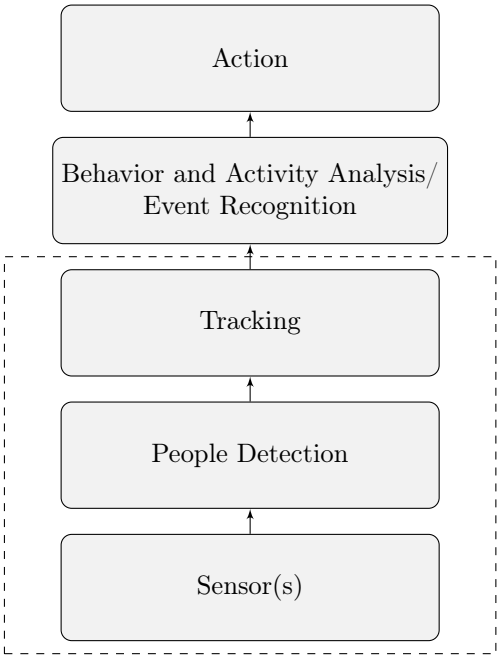


Figure 1.1: Essential components of an intelligent public place surveillance system (having people as targets).

In this thesis, the focus rests on surveillance of people in public places. The objectives of automated surveillance of people in public places are detecting, tracking, and monitoring the activities of people in a public place trying to identify specific events. The main applications of this pertain to public place safety through identification of malicious/hostile individual activities and identification of assistance seeking people automatically. Figure 1.1 depicts a very generic schematic that highlights the major components involved in automated surveillance of people in public places. First, the system perceives the environment using the sensor(s) available in the environment. All people in the monitored area are then detected using the input provided by the sensor(s). The detection is followed by a tracking module which helps capture spatio-temporal information pertaining to each unique person in the monitored area. The spatio-temporal information characterizes the activities of each individual person. Using this information the behavior and activity analysis module infers whether the activities

carried out by each individual digresses from the norm or not, and whether or not their activities conform to any of the events the system is tuned to monitor. In case a sought event is recognized, the system proceeds with action which could be alerting a human operator, sounding an alarm, or sending a robot for intervention/service if a robot is part of the system. An illustrative example is shown in figure 1.2 where a video based surveillance system is used to detect a luggage abandoned by people in a train station and raise an alarm to notify an operator [Porikli 2008]. This system could potentially be coupled with the robotic systems depicted in figure 1.3 (taken from [Jarvis 2008]) for automated luggage inspection and removal. The robots provide a means for a action paving the way for a fully automated active surveillance system (the system of [Porikli 2008] can be considered passive as it has no means for direct intervention). This scenario demonstrates the evident advantages brought up by both fixed and mobile platforms.

Figure 1.2: Automatic abandoned luggage detection [Porikli 2008].

(a) An X-ray inspection robot. (b) A luggage removal robot.

Figure 1.3: Robotic systems for safe inspection and removal of suspicious/abandoned luggage, [Jarvis 2008].

The kind of people surveillance systems just described beforehand are highly demanded in public places like airports, museums, transport stations, *etc*, which are generally wide and complex environments. The first main challenge here is providing optimal surveillance coverage by deploying reasonable number of sensors taking required computational processing, financial expenditure, and surveillance requirement (task, accuracy, robustness, *etc*) into consideration. For example, consider the problem of implementing a people surveillance system in the sample airport terminal shown in figure 1.4. This terminal is a large scale complex environment composed of smaller areas with different properties: narrow passages, wide open areas, very secure areas (near security clearance), *etc*. Depending on the kind of sensors used, the actual number of sensors and configuration required to provide exhaustive coverage would vary. Lets present the pros and cons of, for example, considering the prominent sensors/perceptual systems shown in figure 1.5.

With a classical fixed view camera (figure 1.5a), the number of cameras required to cover all spots will be very high, requiring high bandwidth, processing units, and becoming costly.

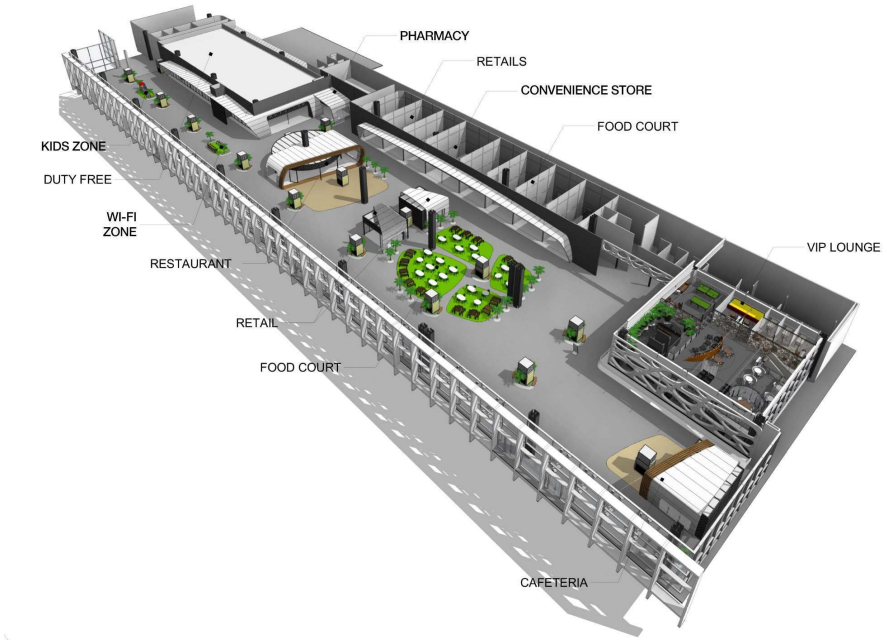


Figure 1.4: 3D floor plan of an airport terminal [San Jose del Cabo Airport].

(a) (b) (c) (d)

Figure 1.5: Examples of some perception sensors/systems. (a) A fixed view wall mounted classical camera; (b) a PTZ camera; (c) a *Ladybug2* camera system that can capture spherical images, and (d) multiple cameras and a movable laser scanner mounted on the mobile robot PR2.

But, as the sensors are stationary, simple and fast algorithms like background subtraction and optical flow could be used to detect moving persons within the camera FOV. Depending on actual sensor configuration, their FOV can cover a wide area—therefore, providing global perception. They can view and track subjects over a broad area for an extended period of time. Additional pitfalls include evident dead-spots that could arise from configuration (placement and number of sensors used), possible occlusions, and their passiveness (inability to change physical position). Apparently, research works based on these sensors are vast in number [Hu 2004, Wang 2013]; they include works that use a single classical camera, network of overlapping [Wang 2013] and/or non-overlapping cameras [Meden 2012, Arsic 2008].

PTZ cameras (figure 1.5b) fixed in the environment, on the other hand, use pan/tilt/zoom functionality to monitor wider areas and zero in on specific individuals, objects, or events. As a result, fewer number of cameras need be used to cover an area with time interleaved coverage with variable resolution. The downside with these cameras is that the camera motion

combined with network latency makes camera modeling, and people detection and tracking difficult [Paillet 2013]. In addition, as long as they are fixed in the environment, their field of view is limited to the visible area covered by all orientations made possible with their actuators. Alternative solutions also include high resolution omnidirectional cameras like the *Ladybug2* camera systems shown in figure 1.5c. This types of cameras can provide 360^o FOV. Though appealing, they are quite expensive, computationally demanding due to high resolution panoramic images, and as long as they are fixed in the environment, they only provide video feed of a pre-fixed area.

(a) (b) (c) (d)

Figure 1.6: Exemplary assistance robotic systems in action. (a) Roboporter experimentally operated by Yasakawa Electric at Kita Kyushu Airport in Japan [Roboporter]. (b) TOOMAS shopping guide [Gross 2009]. (c) T-34, a security robot that nets intruders with spider web spray [T-34]. (d) The Guardrobo D1, from Sohgo Security Services, is designed to patrol office buildings. It can even put out fires [Cnet 2005].

The next consideration is employing a mobile platform, like a mobile robot, with mounted sensors (example in figure 1.5d). In this context, a mobile robot serves two purposes: as a means for action (as discussed previously) but also as a mobile sensor. As a mobile sensor unit, it is generally more suited for surveilling and/or monitoring large areas as this paves a way to reduce the environment structuring and the number of devices needed to cover a given area [Di Paola 2010]. On the other hand, sensors mounted on robots provide localized perception and can pick up details. As a result, robotic based surveillance applications are mostly limited to activities that require close monitoring. They are also suitable for patrolling wide areas owing to their ability to re-position themselves. As a means for action, they are indispensable to realize a complete autonomous intelligent surveillance system, be it for providing service, assistance, or taking counter measures. Figure 1.6 shows examples of using a mobile robot as a means for action in different contexts. In figures 1.6a and 1.6b, it is used to provide assistance by carrying luggage and guiding through a shopping center respectively; and figures 1.6c and 1.6d show experimental mobile systems used as an action means for intruder apprehension and fire extinguishing respectively. The introduction of a mobile robot in public, possibly crowded, environment, actually brings about a new challenge: safe robotic navigation in human occupied environments. It is expected to navigate safely without harming any people in the environment, carrying out its activity reliably in a socially acceptable manner [Sisbot 2007]. This adds more challenges entailing accurate perception (detection and tracking) of people during motion, with the limited computational processing power on-board which is shared by the complete functioning system.

Each of the above mentioned sensor types and configurations have their own advantages and disadvantages. By using any of these configurations cooperatively, it is possible to take benefit of the advantages in each mode. For example, in recent years, researchers have considered surveillance systems that incorporate mobile robots and environment fixed sensors cooperatively, *e.g.*, [Chakravarty 2009, Chia 2009]. These cooperative surveillance systems combine the merits of fixed and mobile perception modes. They acquire global and wide area perception from the

fixed sensors, localized perception and a means for action from the mobile robot. In turn, the robot can take benefit of the global perception from the fixed cameras to better plan its motion to realize safe navigation. This kind of cooperative systems have the potential to lead to more generic surveillance systems as they can handle various scenarios. Cooperation can be achieved in various ways, in a centralized, hierarchical, and decentralized way. To present this systematically, lets consider mono and multi-sensor surveillance configurations.

Mono-sensor Configuration This configuration employs a single sensor and its configuration can be depicted as in figure 1.7, a single sensor directly connected to a processing unit. The sensor can be any one of the sensors discussed in Appendix A capable of providing information for people detection. Commonly, a fixed view camera or PTZ camera either fixed in the environment or mounted on a mobile platform are used. This configuration is very basic and only used to surveil a simple environment. It is mostly used as a basic building block for complex multi-sensor based surveillance systems.

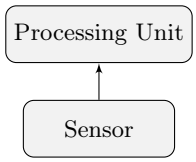


Figure 1.7: Mono-sensor configuration.

Multi-sensor Configuration In this configuration, the surveillance system is composed of many sensors positioned in the environment, possibly fixed with the exception of sensors on a mobile platform. The sensors could be positioned with overlapping FOVs, disjoint FOVs, or a mix of both. In addition to increased coverage, combining different sensor input gives more accurate information, and makes the system less vulnerable to the failure of a single sensor. The following three strategies are widely used cooperation strategies in the literature:

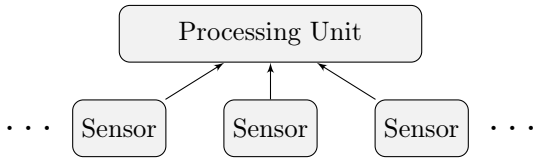


Figure 1.8: Centralized cooperation.

- Centralized Cooperation - in this strategy, data from all sensors are sent to the processing unit. The processing unit then processes the data to infer a global belief on the current state of the environment. The coordination strategy is illustrated in figure 1.8. This strategy is simple to design, and achieves superior state estimation as it uses quite redundant data. Unfortunately, it requires high bandwidth and consequently is not scalable. But, it is ideal for moderate size area requiring fewer sensors for coverage. This cooperation strategy is the most widely used in the literature [Valera 2005].

- Hierarchical Cooperation - in this strategy, the sensors are directly connected to a processing unit which by itself is an intermediate processing unit passing processed data to the central processing unit (figure 1.9). The central processing unit, can alter the perception modalities of subordinate nodes based on processed data from another node for further verification. This strategy minimizes the computation overhead on the central processing unit by off-loading intermediate processing tasks and consequently reduces the bandwidth requirement. A very good example of this cooperation strategy is the simple combined omnidirectional and PTZ camera combination (sometime known as dual camera system), *e.g.*, [Chen 2008, Scotti 2005]. The first processing unit uses the fixed-view

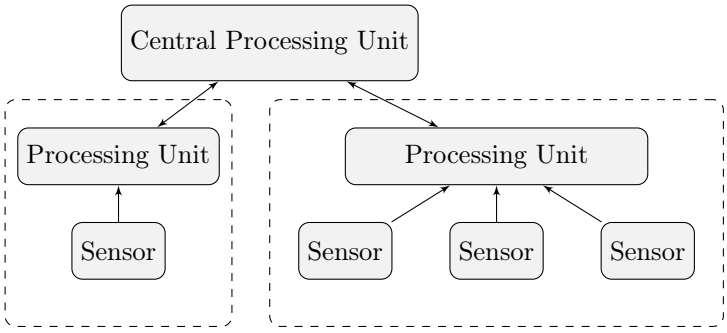


Figure 1.9: Hierarchical cooperation.

camera which has a wider FOV to monitor the entire area, whenever there is an activity in the environment, the node associated with the PTZ camera could be notified so it can orient and zoom in the interesting area to deliver high resolution video of the scenario to the central processing unit. Similarly, in [Chia 2009] three networked wall mounted fixed view cameras and a mobile robot are used to track and follow a target. The target is first detected using the fixed cameras. Once detected, the information is passed onto the robot which navigates to that position and continues to follow the target person. In the vein of Laurent Fit Duval master's internship [Duval 2013], he has investigated this scheme with fixed view and PTZ camera in our research group.

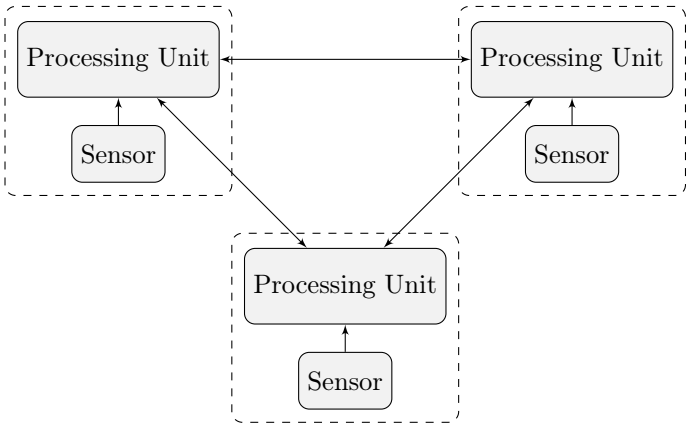


Figure 1.10: Decentralized cooperation.

- Decentralized Cooperation - the decentralized strategy uses autonomous sensor nodes with a processing unit and communication facility that enables communication with other nodes. Each node can work on its own and exchanges information with other nodes to improve its current belief on the environment. This strategy offers modularity, scalability, and fault tolerance. Due to the distributed processing, and minimal intra-nodal communication, it can handle large environments with high number of sensors [Valera 2005]. This configuration is inherently difficult to design and control.

Now, coming back to the airport environment in figure 1.4, clearly, a centralized cooperation scheme for the entire infrastructure is not possible due to bandwidth and high computational

resources entailed. Though utilizing a fully decentralized approach might sound appealing, a mix of the different cooperation schemes presented would be more advantageous. For example, for the wide open areas, a low resolution omnidirectional camera coupled with several PTZ cameras in a hierarchical configuration could be envisaged. Near the security clearance area, a centralized configuration with multiple cameras with overlapping FOV could be used to provide more accurate perception. A mobile robot, on the other hand, decreases the number of sensors deployed by covering dead spots, provides localized high resolution footage of an area whenever required, and provides a means for action either by offering assistance or intervention. It induces flexibility in terms of cooperation strategies with environment fixed sensors: Depending on its location relative to the other sensors, centralized, if overlapping FOV, no cooperation (stand-alone) for dead-spot, decentralized if the robot is located at a hall entrance (each sensor unit will exchange its informations regarding the targets entering/leaving their zone), *etc.* Hence, to fulfill all the requirements stipulated, a multi-sensor surveillance system privileging the different cooperation strategies discussed above should be envisaged.

In summary, robust and fast people detection and tracking are the basic functions of a generic people's surveillance system. A people's surveillance system intended for a wide complex area should envisage a multi-sensor cooperative system with different cooperative modes configured for the specific surveilled areas' perceptual requirements. With this in mind, this thesis first develops a generic visual people detector than can be used from environment fixed or mobile cameras taking both robustness and computational speed into consideration. It then proposes cooperative perception (detection and tracking) modalities using a mobile robot and wall mounted fixed-view cameras suitable for two kind of areas (in a complex environment): moderate sized areas that require increased accuracy, and for areas not covered by the sensors fixed in the environment (dead-spots) with the help of a mobile robot (a currently ongoing investigation). The prospect of human aware robotic navigation using the perceived people's whereabouts is also investigated.

1.2 Scope and Outline of the Thesis

The scope of this thesis falls in the vein of automated surveillance of people in public places spanning surveillance aspects starting from sensor data acquisition to automated people detection and tracking (the dotted rectangular bounding box in figure 1.1). First off, we strongly contend the inclusion of a mobile robot in a public surveillance system. Mobile robots (can) have multiple sensors on-board. As stated before, a mobile robot brings about the following two main vital amenities for any surveillance system.

- Mobile sensor unit - since mobile robots have multiple sensors on-board, they can be considered as a mobile sensor unit. This can reduce required number of sensors to cover an area and helps provide localized perception of an area that needs further verification/inspection. This brings about improved target detection and identification since perceptions from multiple directions (coordinated with mobility) can be obtained to improve the knowledge about the target.

- Means for action - mobile robots are a means for action. They can provide service to the people in the environment, *e.g.*, guidance, assistance (carry luggage). They can also be used for intruder intervention/apprehension and preventive measure implementation in case of an accident.

The addition of a mobile robot comes with additional constraints though. The first relates to the computational resources on-board the robot. The perception modules on-board—which

makes one component of an entire functioning system—should be computationally cheap to levy a reactive robotic system. Second, during navigation, the mobile robot should be able to take the perceived people, in the surrounding, into consideration to realize safe mobility/navigation. Hence, any configuration and/or utilization must address or take these constraints in to consideration prudently. A mobile robot, could be used in any of the three cooperative configurations. It can be used in a decentralized standalone mode processing perceptions on its own and sharing minimal information with other decentralized nodes; it can be employed in a hierarchical scheme liaised with other nodes through a central supervision unit; or it can be used in a centralized manner making the most out of all perceptions on-board and from the environment fixed sensors.

In this thesis, we will consider the use of a mobile robot in two configurations. The first configuration is a centralized mode in which the sensor data from the mobile robot are fused with sensor data from environment fixed cameras to implement a multi-person detection and tracking functionality. This centralized cooperation results in improved perception and makes it ideal for moderate sized areas that require increased tracking accuracy (for example, near security zones in the airport terminal shown in figure 1.4). In the second configuration, for which only a future prospect is presented in this manuscript, the mobile robot is used as a standalone unit to detect and track people in its vicinity. In figure 1.4, this could be used to cover all spaces not covered by environment fixed sensors. This mode by itself fuses data from multiple sensors on-board the robot in a centralized manner, but it can be seen as a decentralized (self-contained) perceptual component with respect to the entire airport surveillance system. In both configurations, all constraints by the mobile robot are carefully dealt with by first focusing on computation aspects during detector development and second by using the perceived people information within the robot navigation scheme to realize safe robotic navigation. All in all, the work carried out in this thesis are generic and serves some of the needs in the public place surveillance context presented in section 1.1 which are quite challenging and not so common in the literature. The focus is bestowed on two aspects: people detection and cooperative tracking. Both of these aspects relate to the literature in their corresponding realms.

1.2.1 Automated People Detection

The first part of this thesis deals with automated people detection using a visual camera. The literature in visual people detection is overwhelming, various researchers proposing many different detection approaches. In our investigation, we give extra focus on computation time so as to develop a detector that not only has acceptable detection performance but is also fast. We do this by focusing on developing a visual people detector that optimizes over detection performance and computation time explicitly without any assumption on sensor motion. Another important development constraint taken into consideration is that this detector should be equally applicable on a visual sensor that is either fixed in the environment or mounted on a mobile platform (no assumption on sensor motion).

The explicit computation time consideration is of paramount importance as we are considering a mobile robot which has to share its limited computation resources with other computations involved to maintain a functioning autonomy. For example, considering the famous Dalal and Triggs [Dalal 2005] people detector by default off-the-shelf, would harness 0.2 fps on a 640×480 image using PIII single core machine running at $800Mhz$. Similarly, in the literature, many researchers report significant proportion of computation time taken by people detection modules, for example, $> 90\%$ (6 seconds per frame) in [Ess 2010], and $50\% - 66.7\%$ in [Choi 2011], of the complete time taken by the perceptual system, on each iteration, leading to reduced to impractical robot reactivity. This has even been a main reason, in addition to detection accuracy, that forced some researchers to do research experiments via offline evaluations, *e.g.*, [Volkhardt 2013], other

very expensive motion capture devices with marker tagged people, *e.g.*, [Pandey 2010], as an example, when doing experiments that require people perception in a robot's vicinity.

1.2.2 Cooperative Multi-person Tracking

The scope of the second part of the thesis rests on tracking multiple people in a monitored environment using a cooperative multi-sensor surveillance configuration. In this part, two different tracking systems are realized. The first relates to a centralized cooperative configuration using sensors fixed in the environment and mounted on a mobile robot (presented in full detail). In the second system, a stand-alone multi-person tracker using sensors solely on-board the mobile robot is considered (only an excerpt provided in this thesis). This system is aimed at detecting and tracking people in areas not covered by wall mounted sensors like dead-spots. This system can actually be seen as a decentralized perception system with respect to an entire surveillance system (for example, as might be required in the airport terminal of figure 1.4). In both cases, the mobile robot makes use of the perceptions for safe navigation in crowds beyond the surveillance objective.

1.3 Contributions and Organization

1.3.1 Contributions

This thesis document is presented divided into two parts. The contributions made in each part are detailed herewith.

The contributions made by the first part of this thesis are in the vein of visual people detection. First, a comprehensive review of the state-of-the-art in visual people detection is presented. Then, we present a novel mathematical formulation based on Binary Integer Programming (BIP) for feature selection taking both computation time and detection performance into consideration. This notion of explicit computation time consideration and optimization in the detector characterization (learning) is rarely considered in the literature. This framework is initially validated using Histogram of Oriented Gradient (HOG) [Dalal 2005] features. It is then extended to incorporate heterogeneous pool of features and compared against alternative heterogeneous feature mining techniques inspired from the literature. In all cases, the developed detectors are thoroughly evaluated using three datasets—(1) a proprietary dataset compiled using images taken by the *Ladybug2* spherical camera, (2) the INRIA public dataset [Dalal 2005], and (3) the Caltech dataset [Dollár 2012]) datasets—and compared with the state-of-the-art.

The second part of the thesis deals with cooperative multi-target tracking. Similar to part I, it starts out with a review of the state-of-the-art in multi-target tracking and multi-sensor cooperation schemes. It then proposes and validates a centralized cooperative framework and data fusion scheme between wall mounted fixed view cameras and sensors embedded on a mobile robot to track multiple passers-by in a surveilled area. The improvements brought upon by the cooperative fusion are thoroughly evaluated. It then deploys the developed perceptual functionalities on an actual robot platform demonstrating, (i) how the robot makes use of the perceived information to realize safe navigation in human occupied environment, and (ii) how the proposed optimized detection (in part I of the thesis) improves the reactivity of the mobile robot. This mode is applicable to a moderate sized area where improved accuracy is required. Additionally, the same cooperative data fusion scheme is considered in a self-contained mode using a novel high resolution spherical camera (the *Ladybug2* camera) and laser range

finder on-boarded on a mobile robot. We present a short overview of future prospects in this vein.

In the course of this thesis development, the following list of publications have been realized:

International publications

Journals

- [Mekonnen 2013c] *A. A. Mekonnen*, F. Lerasle, A. Herbulot, Cooperative Passers-by Tracking with a Mobile Robot and External Cameras, Computer Vision and Image Understanding (CVIU'13'), vol. 117, no. 10, Pages 1229-1244, 2013.
- [Zuriarrain 2013] I. Zuriarrain, *A. A. Mekonnen*, F. Lerasle, N. Arana, Tracking-by-detection of multiple persons by a resample-move particle filter, Machine Vision and Applications (MVA'13), vol. 24, no. 8, pages 1751–1765, 2013.

Conferences

- [Mekonnen 2014a] *A. A. Mekonnen*, F Lerasle and A Herbulot. People Detection with Heterogeneous Features and Explicit Optimization on Computation Time, International Conference on Pattern Recognition (ICPR'14), Stockholm (Sweden), August 2014.
- [Mekonnen 2013a] *A. A. Mekonnen*, C. Briand, F. Lerasle, A. Herbulot, Fast HOG based Person Detection devoted to a Mobile Robot with a Spherical Camera, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13), Tokyo (Japan), November 2013.
- [Mekonnen 2013e] *A. A. Mekonnen*, F. Lerasle, A. Herbulot, Pareto-Front Analysis and AdaBoost for Person Detection using Heterogeneity Features, IEEE International Conference on Systems, Man, and Cybernetics (SMC'13), Manchester (UK), October 2013.
- [Mekonnen 2013f] *A. A. Mekonnen*, F. Lerasle, A. Herbulot, Person Detection with a Computation Time Weighted AdaBoost and Heterogeneous Pool of Features, Advanced Concepts in Intelligent Vision Systems (ACIVS'13), Poznan (Poland), October 2013.
- [Mekonnen 2013d] *A. A. Mekonnen*, F. Lerasle, A. Herbulot, External Cameras and a Mobile Robot for Enhanced Multi-person Tracking, International Conference on Computer Vision Theory and Applications (VISAPP'13), Barcelona (Spain), February 2013.
- [Mekonnen 2011] *A. A. Mekonnen*, F. Lerasle, I. Zuriarrain, Multi-modal Person Detection and Tracking from a Mobile Robot in a Crowded Environment , International Conference on Computer Vision Theory and Applications (VISAPP'11), Algarve (Portugal), March 2011.

National publications

- [Mekonnen 2014b]*A. A. Mekonnen*, A. Herbulot, F. Lerasle, C. Briand, Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU, 19ième congrés francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'14), Rouen (France), Juillet 2014.
- [Mekonnen 2013b]*A. A. Mekonnen*, A. Herbulot, F. Lerasle, Coopération entre perception déportée et embarquée sur un robot guide pour l'aide à sa navigation, dans Revue d'Intelligence Artificielle (RIA'13), volume 27, pages 65-93, no. 1, 2013.
- [Mekonnen 2012] *A. A. Mekonnen*, F. Lerasle, A. Herbulot, A. Coustou, Coopération entre un robot mobile et des caméras d'ambiance pour le suivi multi-personnes, 18ième congrés francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'12), Lyon (France), Janvier 2012. [Article sélectionné pour parution dans le journal RIA.]

1.3.2 Manuscript Organization

To provide a complete and systematic account of the work carried out in this thesis, the rest of this manuscript is organized as follows:

- **Part I - Automated People Detection**

  The first part of the thesis deals with automated people detection from a visual sensor. The part by itself is divided into three chapters. It begins by presenting the trends, modes, and different considerations of automated people detection in the literature in chapter 2. This chapter discusses the state-of-the-art in people detection generally with emphasis on vision based approaches. It is used as a stepping stone to put the contributions made in subsequent chapters (3 and 4) in perspective.

  Chapter 3 presents an optimized HOG based person detector show casing a novel feature selection framework based on Binary Integer Programming (BIP). A detailed formulation of this optimization framework and its application to HOG feature based optimized detector learning is presented here. This framework is extended to various heterogeneous features with a thorough evaluation and comparison with the state-of-the-art in chapter 4.

- **Part II - Cooperative Perception for Tracking People**

  The second part of the thesis focuses on cooperative multi-target tracking of people in a surveilled area. Similar to Part I, it begins with a presentation of the state-of-the-art in cooperative data fusion strategies and multi-target tracking in chapter 5. Again, this chapter is used to highlight the contributions made in subsequent chapters in context with the literature.

  In chapter 6, a centralized cooperative perception strategy between fixed-view cameras fixed in the environment and sensor(s) on a mobile robot is discussed. The chapter also presents implementation details and experiments carried out with the mobile robot to demonstrate the advantages of this proposed framework.

- **Conclusions and Future Prospects**

  Finally, the manuscript finalizes with a summary of our contributions along with conclusive remarks and future prospects in chapter 8.

Part I

AUTOMATED PEOPLE DETECTION

Chapter 2

Trends, Modes, and Considerations

Contents

2.1 Introduction **15**
2.2 Trends and Modes in the Literature **17**
2.2.1 Sensors 17
2.2.2 Vision based Detection 19
2.2.3 Multi-modal Approaches 28
2.2.4 Discussions 29
2.3 Visual Datasets **32**
2.3.1 Ladybug Dataset 32
2.3.2 INRIA Person Dataset 33
2.3.3 Caltech Pedestrian Dataset 34
2.4 Evaluation Metrics **35**
2.4.1 PW Evaluation: Detector Error Tradeoff (DET) 36
2.4.2 PW Evaluation: Precision-Recall 36
2.4.3 Full Image Evaluation 36
2.5 Summary **37**

2.1 Introduction

In modern era computer vision is playing a significant role in automated object perception; one such thriving role is automated people detection. People detection is one of the prominent prob-

lems considered in computer vision. It has a vast pool of applications spanning many research domains. Prominent areas where automated people detection is invaluable are: Human-Robot Interaction, Human-Computer Interaction, Pedestrian Protection Systems (part of Advanced Driver Assistance Systems), Video Surveillance, and Automated Image Indexing and Management.

Automated people detection involves perceiving the whereabouts of people in the information of a scene captured by a sensor. Depending on the mode of the sensor, this can mean localizing the accurate 3D position or rough 2D position of each person in the scene. Unfortunately, person detection is by far one of the most challenging tasks in computer vision mainly because of the following reasons:-

- *Physical variation of people*: People's appearance varies greatly. People exhibit different body sizes (physical variation), and different color and texture appearances (as a result of the cloths they wear).
- *Body deformations due to articulation*: For a detection system that depends on the shape of a person, body shape deformations can adversely affect the detection system.
- *Illumination variation*: For a detection system that depends on the lighting condition, varying illuminations and shadings in different environments can affect the detection.
- *Viewpoint change*: Depending from which angle people are viewed, they can yield different shapes with varying aspect ratios.
- *Background clutter*: Sometimes background structures in the scene exhibit similar structure and shape as that of a person, making distinction difficult.
- *Occlusions*: Sometimes people are partially or completely occluded by, things they are carrying, overlaps with other people, or by structures in the environment, hence, making successful detection very difficult.
- *Sensor limitations*: In robotic context, most embedded sensors have short fields of view and they are usually mobile, making the detection task difficult.
- *Computational constraints*: Techniques and methods that achieve state-of-the-arts detection usually require a lot of computation time compared to trivial person detection method. This poses a challenge in real-time systems (*e.g.*, robotics, automotive applications) where it is required to have reactive response acceptable by humans. Balancing detection performance with computational requirement adds to the challenge faced in people detection.

Different methods have thus far been proposed by various researchers in the hopes of overcoming the above outlined challenges. The main objective of this chapter is to recap the gist of the different methods proposed for automated people detection in the literature. The chapter will start by highlighting the general trends (section 2.2) and will continue with an in-depth focus on vision based detection methods (section 2.2.2) and a brief discussion on the presented approaches in section 2.2.4. Finally, it will conclude with a presentation of different datasets (section 2.3) and evaluation metrics (section 2.4) that will be used in subsequent chapters of this part of the thesis. This will make the contributions set forth in chapters 3 and 4 apparent and reduce any redundant information between them.

In the literature, two main distinctions about people detection can be made. The first is pedestrian detection which aims at detecting people that usually exhibit more regularities in pose and appearance (upright postures) in outdoor scenes. And the second, generic people

detection where people exhibit much larger pose variations in unconstrained environments like homes, malls, and other indoor environments. Clearly, pedestrian detection is more tractable than generic people detection, but it also faces further complications because of environment factors (*e.g.*, fog, rain, *etc*) and sensor motion when mounted on a vehicle. In this work, generic people detection techniques that try to handle high variations in articulation are considered as tackling pose estimation problems. And hence, in subsequent parts, we consider both pedestrian and generic people detection with reduced articulation variation (as found in video surveillance applications) as people detection without no loss of generality.

2.2 Trends and Modes in the Literature

Undoubtedly, automated people detection is a very important research area with prominent applications in video surveillance, robotics, human-computer interaction, and image indexing. Generally speaking, detection makes one part of the perception pipeline, traditionally followed by tracking. In the literature, various sensors ranging from a simple binary switch that measures the presence of a person to complex sensors that capture 3D cloud of the environment have been used. In video surveillance and driver assistance systems visual camera based detectors dominate. On the other hand, in robotics, as most robotic platforms are equipped with several sensors, the trend is to use two or more different sensors to make a multi-modal detector.

This section begins by briefly presenting the different sensors that have thus far been used for people detection. It then focuses on vision based detectors as these take up the lions share in the literature. Furthermore, multi-modal approaches are discussed briefly. Finally, the section concludes with a discussion that highlights the important points of the presented methods and puts the contributions made in chapter 3 and 4 into context.

2.2.1 Sensors

In the literature, a variety of sensors have been employed for automated people detection. All these sensors can be boldly categorized into active and passive sensors. Active sensors work by radiating some sort of radiation on to the object/scene and provide measurement information inferred from the reflected radiation. On the other hand, passive sensors provide measurement information that is directly obtained from the levels of energy that are naturally emitted, reflected, or transmitted by the object/scene. Putting budgetary issues aside, the specific choice to use a distinct type of sensor is motivated by the application context: required information and environment interference. Table 2.1 summarizes the different sensors used for detecting people along with the mostly used detection approaches. A succinct presentation of each sensor can be found in appendix A (an extensive survey is provided in [Teixeira 2010]).

Generally, visible spectrum cameras (also referred to as classical cameras) are the most widely used sensors for people detection. These sensors capture very informative data covering wide spatial area, with color and texture information of the scene. They are also quite versatile and cheap. Omnidirectional versions—cameras that can capture more than 180^o of a scene either by using special lenses, reflective mirrors, or multiple camera configurations—are very useful due to their spatial coverage. As a result of their unparalleled utilization and advantages, we will focus the presentation in the next sections and the investigations in subsequent chapters on visible spectrum cameras including a spherical omnidirectional camera.

Sensor	Active/ Passive	Detection Technique
Classical Camera (visible spectrum)	Passive	various methods described in section 2.2.2; relevant surveys in [Dollár 2012, Gerónimo 2010a].
Thermal Camera	Passive	based on human heat signature which stands out; image segmentation based on thresholding, noise filtering, and morphological operations, *e.g.*, [Correa 2012, Treptow 2005].
Stereo Camera pair	Passive	3D blob segmentation for candidate generation, further verification using 2D image, *e.g.*, [Muñoz-Salinas 2007, Gerónimo 2010b].
Structure Light based RGB+D (*e.g.*, Kinect)	Active	3D blob segmentation [Salas 2011]; 3D features with statistically learned classifier [Spinello 2011].
Microphone(s)	Passive	localizing direction of sound source (requires at least two microphones), assuming a human speaker with no other interference, *e.g.*, [Brückmann 2006, Bennewitz 2005].
Lidar (2D/3D)	Active	2D laser range finders: segmentation based on geometrical features in an ad-hoc fashion [Xavier 2005] or using statistical learning algorithms [Arras 2007]. flash lidar camera: 3D blob segmentation [Ikemura 2011].
Sonar	Active	segmenting the scan taking geometric constraints into consideration (based on expected scan profiles of people at the mounted height) [Martin 2006].
Radar	Active	segmentation via clustering [Milch 2001], background subtraction for moving targets [Zetik 2006].
RFID reader	Active	detection of RFID tags worn by people [Germa 2010].

Table 2.1: Summary of different sensors with associated characteristic approaches for people detection.

2.2.2 Vision based Detection

2.2.2.1 Overview

Vast majority of works on person detection utilize visible spectrum cameras. People detection based on visible spectrum cameras is here collectively referred to as vision based people detection or visual people detection.

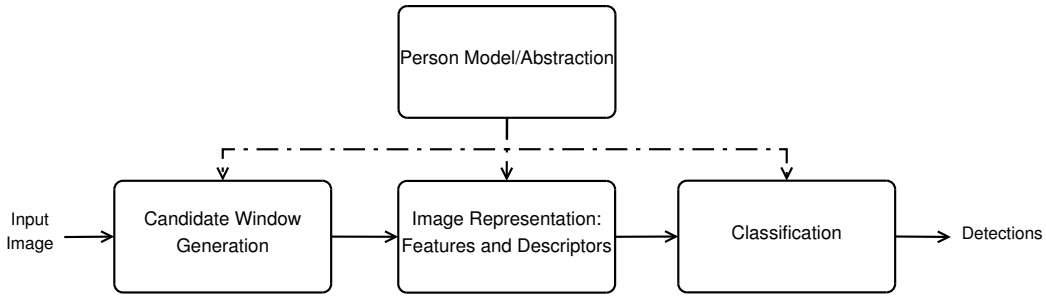


Figure 2.1: Important components of a vision based person detector.

All methods in the literature more or less adhere to the generic scheme depicted in figure 2.1. For a given input image, first possible candidate windows are hypothesized. Using the person model adopted, the original raw image input is transformed into a convenient format by extracting certain types of features that capture specific cues and rearranging them into a prior fixed descriptor. Finally, each hypothesis is labeled as either a person or not using a learned classification rule. Though not shown in the figure, there is usually a last post-processing step in the form of *Non-Maximal Suppression*(NMS). Its purpose is to merge multiple detections that may arise from the same person into one. Two main approaches in the literature are the *Mean Shift* (MS) mode estimation [Dalal 2006a] and a *Pairwise Max* (PM) [Felzenszwalb 2010b] suppression. PM works by discarding the less confident of pair of overlapping detections while MS, as the name implies, uses mean shift to estimate the mode of the detections. The scheme shown in figure 2.1 shows the flow used during detection. The types of features, descriptors, classifiers along with the exact person model employed is a detector design choice. But, the actual subset of features/descriptors to use and the exact classifier parameters are determined via a training, also called learning, phase using a training dataset that contains positive and negative instances. Figure 2.2 shows an illustration of the people detector learning procedure used by [Dalal 2005].

2.2.2.2 Person Models/Abstractions

The general people detection pipeline (shown in figure 2.1) has three main components, namely: candidate window generation, image representation, and classification. All these blocks actually make use of an underlying abstraction or model that dictates how a person is represented. For example, weather to look for full human bodies or to look for human body parts and bundle them to infer presence of a person is determined by the specific abstraction/model employed. In the literature, two main distinctions can be made: Implicit and Explicit methods. The implicit methods do not use cues that are specific to humans, rather use other information (motion or deviation from norm) as an indicator. On the hand, explicit methods do actually use cues that are specific to people. Figure 2.3 depicts a taxonomy of people detection methods that vary based on the exact abstraction/model used.

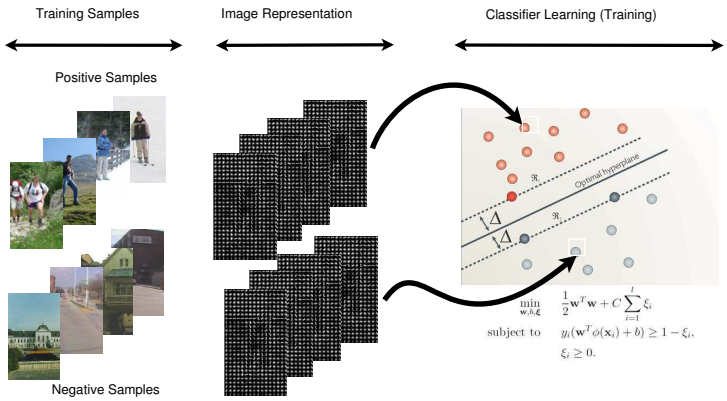


Figure 2.2: Illustration of an exemplar people detector learning scheme.

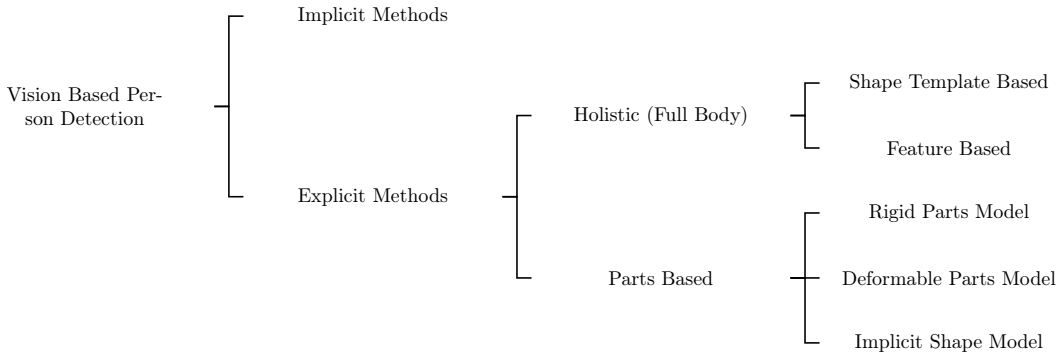


Figure 2.3: Taxonomy of visual person detection methods.

Implicit Methods These methods detect people by considering their deviation from the background. They are called implicit because they do not explicitly infer the presence of people rather they segment out foreground objects and label them as people if they satisfy the aspect ratio of an average person. Two prevalent techniques in this category are background subtraction [Piccardi 2004] and optical flow analysis [Beauchemin 1995]. Background subtraction techniques model the background and label each pixel of a new frame as a foreground or background. Pixels from people in the environment can thus be labeled as foreground based on their deviations from the background. Finally, foreground pixels are connected and filtered using morphological operations to form blobs which are considered as people or not based on their aspect ratio [Ekinci 2003]. The downside is that this technique works only from static cameras and it can easily be fooled by foreground objects that might have comparable aspect ratios to humans. On the hand, optical flow based techniques trace motion of each pixel in consequent frames and thus are able to segment out moving objects. Similarly, moving people can be detected with an aspect ratio constraint without any explicit model.

Direct people detection with either background subtraction technique or optical flow segmentation lacks explicit considerations and hence is prone to mistakes. It is better to use these techniques as an initial hypothesis generation schemes for subsequent verification with explicit methods, *e.g.*, [Haga 2004, Toth 2003].

Explicit Methods Contrary to implicit methods, explicit methods make use of some kind of model or abstraction that captures salient discriminant attributes that distinguish people from other information in an image. The model's or abstraction's exact configuration and/or parameters are determined using positive and negative training examples. A vast majority of works in the literature on image based people detection fall in this category [Dollár 2012, Gerónimo 2010a, Enzweiler 2009]. Explicit methods can be further divided into *holistic (full body or monolithic) approaches* and *parts based approaches*. The holistic approaches search a given image for a full human body based on a full body abstraction of a person. On the other hand, parts based approaches try to aggregate evidence of a person's existence by using a part based human body model and looking for these body parts.

Holistic (full body or monolithic) approaches: These approaches consider a person as a whole indivisible object. One such method is a shape template based approach for person detection [Gavrila 2000, Gavrila 1999, Broggi 2000, Broggi 2006]. In this approach, a set of people shape templates are constructed from training positive examples. Then people are detected by scanning input images at different scales looking for similar shape structures as any of the templates. For example, Gavrilla [Gavrila 2000, Gavrila 1999] constructed a hierarchy of binary shape templates using k-means like clustering. During detection, distance transform in conjunction with Chamfer distance and a threshold is used to detect people in the image traversing the template hierarchy in a coarse-to-fine paradigm. Evidently, huge number of templates are required to capture the variations in articulations. This increases the processing time significantly during detection. It has been pointed out silhouette matching methods are not applicable in general as standalone techniques and rather require an extra appearance-based verification step for acceptable performance [Gerónimo 2010a].

Another method that has been considered extensively is the feature based approach for full body detection. In this category, a full body person model is learned using features extracted from positive and negative candidate window training examples by employing one of the discriminant classifiers discussed in section 2.2.2.5. This learned model is then used to label candidate windows generated from the input image, with the help of one of the candidate window generation techniques discussed in section 2.2.2.3, as people or not. An extensive discussion of the different features used in the literature is presented in section 2.2.2.4. Briefly speaking, the features used range from the simplest one that considers raw image pixel values to complicated descriptors like the Histogram of Oriented Gradients (HOG) constructed by performing a spatial binning over image gradient orientations. The representation is not restricted to using homogeneous features, but rather heterogeneous pool of features could be considered to incorporate complementary information for improved detection performance. The pioneer in this paradigm is the works of Papageorgiou *et al.* [Papageorgiou 2000] with Haar like features and a holistic person detector learned using an SVM classifier. The work of Dalal and Triggs [Dalal 2005] which introduced and used gradient based features called HOG features was next in line to set the performance bar high. To date, HOG is the most discriminant feature, and in fact, a majority of detectors proposed hence-after make use of HOG or its variant one way or another [Dollár 2012]. Recent techniques that go beyond Dalal and Triggs in this full-body approach utilize heterogeneous pool of features [Walk 2010, Hussain 2010, Dollár 2009, Wang 2009, Wojek 2008].

In general, holistic approaches are quite appealing due to their simplistic abstraction, straight forward model training, and, compared to parts-based approaches, reduced computation time during detection. On the other hand, as they are trained on up-right persons, they are greatly affected by non-standard poses (articulations) and partial visibility due to occlusion or partially being out of camera field of view.

Parts based approaches: Contrary to holistic abstraction that tries to model a person as one indivisible object, parts based approaches rely on detecting different parts of a body–either explicitly looking for a head, torso, arms, and legs or looking for implicit dividends–to detect a person. Broadly speaking, the underlying principle of the parts-based approaches in the literature can be attributed to the *pictorial structure model* [Fischler 1973]. A pictorial structure model for an object defines an object as a collection of parts with connections between certain pairs of parts [Fischler 1973, Felzenszwalb 2005].

One variant of this approach considers rigid parts model that carry semantic information corresponding to anatomical human parts. For example: Mohan *et al.* [Mohan 2001] presented a parts based person detector considering head, left and right arms, and the legs as constituent parts. In their work, first person's body parts are independently detected in an image. The final score is computed by applying a classification step with a linear classifier trained using rigid geometric constraints and part confidence scores from the samples in the training set. Similarly, Mikolajczyk *et al.* [Mikolajczyk 2004] considered a parts based people detector based on a probabilistic assembly of robust part detectors. The joint probabilistic geometrical body parts relation is learned from a training data. The considered parts include frontal head, frontal face, profile head, profile face, frontal upper body, profile upper body, and legs. Wu and Nevatia [Wu 2005] presented a framework for detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. They define a joint image likelihood function for multiple, possibly inter-occluded humans. A missing part is explained as the missing detection of the part detector or occlusion by other objects.

The downside with anatomically associated parts-model is that missing parts, as a result of partial occlusion, affect the overall likelihood of the composite model. To alleviate this, Felzenszwalb *et al.* [Felzenszwalb 2010b] proposed a deformable parts model that selects parts purely based on their visual saliency–discovering model parts in an unsupervised manner–rather than relying on semantic information. The actual parts are not specified a priori, but given labeled bounding boxes for the full body and number of parts, their algorithm selects salient parts from the training data through an iterative optimization with associated deformation map. Similar problem, learning parts based on visual saliency from examples, has also been tackled by Dollár *et al.* using Multiple Component Learning (MCL) [Dollár 2008].

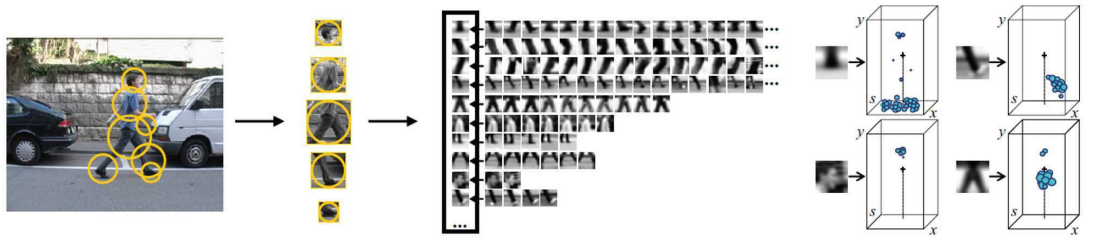


Figure 2.4: Implicit Shape Model. During training, local features are extracted around interest points and clustered to form an appearance codebook. For each codebook entry, a spatial occurrence distribution is learned and stored in non-parametric form (as a list of occurrences), from [Leibe 2008].

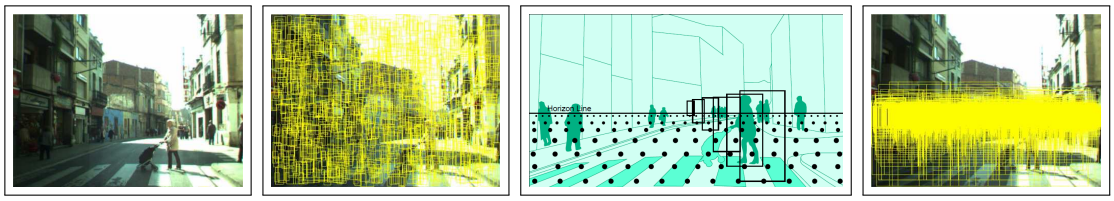
A somewhat differing popular parts-based approach is the *Implicit Shape Model* [Leibe 2008]. Leibe *et al.* represent an object using a codebook of visual words, sampled using interest point detectors from training images and clustered, with learned spatial information with respect to the centroid of a person's bounding box. During detection, image patches found with interest point detectors are matched to the words in the codebook casting a possible vote for object centroid

based on the learned spatial information; persons are detected by determining the local maxima in this voting space. This also avoids the need for a candidate window generating step as it is entirely based on interest point detectors. Illustrations of this method are shown in figure 2.4. This approach builds the codebook and does the inference solely based on the training examples without any assumption on the number of parts or their supervised specification. It captures intra-class variability automatically and is robust to partial occlusions.

In general, parts based approaches are better suited for person detection thanks to their ability to better deal with partial occlusions, view point changes, and pose variations because of articulation. Associating parts to anatomically equivalent human components eases the abstraction, but suffers with missing part components [Mohan 2001, Mikolajczyk 2004]. Robust models can be obtained using generic parts that do not carry any semantic information and are selected purely by their visual saliency [Felzenszwalb 2010b, Leibe 2008, Dollár 2008]. This also avoids the overwhelming task of manually annotating each part and simplifies detection of other objects where deciding semantic parts can be ambiguous and/or subjective. However, the above advantages come at the expense of complex involved training and higher computation time during detection. Parts based methods also perform poorly with lower resolution images as the parts require ample spatial support for robustness; a mechanism could be put in place to use, a parts method when entailed resolution is guaranteed and a holistic method otherwise [Park 2010].

2.2.2.3 Candidate Window Generation

The candidate window generation step addresses the exact technique used to generate plausible person containing hypothesis that need further verification in the input image. In the literature, three main trends are observed: A brute force approach, geometric considerations, and some form of attentional mechanism to segment out interesting position. The easiest and most abundantly used is the brute force approach which is commonly referred to as the *sliding window mode*. In the sliding window mode, candidate windows with a fixed aspect ratio are sampled at all positions and scales of the input image–a brute force approach, figure 2.5b. It neither requires any prior knowledge nor makes any assumption about the scene and camera. This mode is advantageous as no position within the image gets left untested, but its nature lends to increased computation time due to possible scanning of unnecessary locations, *e.g.*, parts of the image corresponding to high levels or ceilings with cameras mounted on mobile robots or vehicles.



(a) (b) (c) (d)

Figure 2.5: In (a) the original image is shown. (b) shows 0.1% of all candidate windows generated using the sliding window mode. (c) depicts illustration of the flat world assumption (geometric constraint) along with how the candidate windows are placed, and (d) shows 5% of the total candidate windows generated this way. (Images taken from [Gerónimo 2010b].)

If the geometric relationship between the mounted camera and the ground is known or can be automatically estimated, it can be used to reduce the total number of candidate windows generated significantly. Assuming a flat world and a calibrated camera, only candidate windows in the image plane corresponding to projections of real world candidate windows placed on the

ground are passed onto the next stage. This geometric consideration will decrease the total number of hypothesis drastically; all hypothesis on high walls/ceilings, in indoor environments, and on the sky, in outdoor environments, are discarded (see figure 2.5c and 2.5d). When dealing with a camera mounted on a moving vehicle, different algorithms that estimate the ground plane and update the model dynamically can be used [Gerónimo 2010b].

Another alternative is some kind of attentional mechanism to identify and then segment potential hypothesis for further testing as a candidate window. If the camera is static, the implicit person detection methods discussed in section 2.2.2.2, namely background subtraction and optical flow, can be used as a candidate window generation mechanisms. It is worth mentioning here that any attentional mechanism that can discard a proportion of the negative windows without missing any person and yet bears lessened computation time compared to the actual classifier can be used, for it will eventually decrease the overall detection time. As a result, different authors have tried to use simple cues that fulfill the aforementioned conditions. Example cues include, color contrast, edge density, superpixel straddling, color symmetry, and edge symmetry as discussed in [Alexe 2010, Paleček 2012]; additional cues, namely verticality, and dominant orientation, are also proposed in Palecek *et al.* [Paleček 2012]. In the robotics community where laser range finders (LRF) are common, it is common to use the laser information, which is very easy and fast to process, to constrain the search space for possible hypothesis [Schiele 2009].

2.2.2.4 Image Representation: Features and Descriptors

The challenges faced in visual people detection are numerous. Using individual pixel values from the input image solely leads to poor detection performance and generalization as individual points do not convey any global information about the appearance of people. Simply by looking at a group of neighboring points useful information and insights about the underlying object can be obtained. For example, by looking at the immediate neighbors of a pixel, the magnitude and orientation of the spatial appearance variation of the underlying object at that point can be determined. Further widening the support region would enable us to capture the edge structural profile of the object. Hence, modified representations of the image are vital for improved image interpretations. Any information extracted in such a way is termed as *a feature*. Features enable us to capture the essence of the underlying scene by extracting meaningful information from a group of data points (pixels). Different features capture differing facets of the underlying scene and careful feature choice plays an important role on the detection performance. In fact, robust image representation and discriminative learning algorithms are the key to recent advances in people detection [Dollár 2012, Gerónimo 2010a].

In the literature, interesting and salient points in an image are referred to as feature points. Usually, a feature point is associated with a descriptor constructed by taking all features in the local region surrounding the feature point and arranging them in a specified manner to make a feature descriptor. This distinction is apparent in the literature in image indexing and matching where sparse representation is commonly used. However, in people detection where dense sampling is dominantly employed, the distinction between a feature point and descriptor fades; each point in the image is treated as a feature point and has an associated descriptor. Consequently, in this work, a feature point and its associated feature descriptor are collectively referred to as just *a feature*.

Early success in people detection was achieved using rudimentary **Haar like features** inspired by Haar Wavelets [Papageorgiou 2000, Lienhart 2002, Viola 2005]. These over-complete family of features compute the summed intensity difference between different regions in a fast and simple way–sum of pixels spanned by the white region minus that of the black region. The feature values are computed efficiently with the help of integral images [Viola 2004]. The fea-

tures capture change in local intensity along different directions. As the number of Haar features that can be computed in a fixed window is high in number (over-complete features in terms of position and scale), a few discriminant ones should be selected using one of the boosting classifier variants [Schapire 2003]. As these features capture region intensity differences, their descriptive power is limited especially considering the distinctive boundary of peoples' figures which can better be captured using edges. This intuition led to the adoption of gradient based features. One such feature is the **Edge Orientation Histogram (EOH)** originally proposed for face detection [Levi 2004]. These features represent ratios of gradients computed from edge orientations histograms. Within a given overlaid region, first gradients are computed. Then, a gradient histogram is built by quantizing the gradient orientations. Finally, the ratios of each histogram bin with one another makes up individual features. These features not only maintain invariance to global illumination changes, but also capture geometric properties and have been shown to improve detection especially compared to Haar like features [Gerónimo 2007].

Another gradient based feature that has proved to be outstanding is the **Histogram of Oriented Gradients (HOG)** [Dalal 2005]. HOG features are extracted first by computing the gradient, then by constructing a histogram weighted by the gradient magnitude in an atomic region called *a cell*. Histograms of neighboring cells are grouped into a single block, cross-normalized and concatenated to give a feature vector per block. Dalal and Triggs [Dalal 2005] concatenated all block histograms inside a candidate window to generate one high dimensional feature descriptor. However, Zhu *et al.* [Zhu 2006] have also shown marginally comparable detection performance can be achieved using a subset of variable sized HOG blocks selected and combined in a boosting framework. To date, HOG is the most discriminant feature and no other single feature has been able to supersede it [Dollár 2012, Gerónimo 2010a]; state-of-the-art results obtained with both holistic [Dollár 2012] and parts-based [Felzenszwalb 2010b] abstractions use HOG or some form of its variant.

Recently, variants of **Local Binary Pattern (LBP)** features have been burgeoning in people detection. Local Binary Patterns were initially proposed as a texture characterization features [Ojala 1996]. The basic idea is to calculate a power two modulated integer label for each pixel by thresholding neighboring pixels by the central pixel going around uniformly. The thresholding step considers relative intensity values making the feature illumination and contrast invariant. Texture patterns with different spatial support could be captured by varying the neighborhood radius and sampled points. The final features could be computed by simply constructing a histogram in a rectangular region, and doing this for all possible rectangular regions inside the candidate window giving rise to over-complete feature set, or by building a high dimensional feature descriptor just like HOG [Mu 2008, Satpathy 2013]. Variants of LBP features proposed in the literature include Non Redundant Local Binary Patterns (NRLBP) [Mu 2008], Discriminative Robust Local Binary Pattern (DRLBP) [Satpathy 2013], and Cell Structured Local Binary Patterns (CellLBP) [Wang 2009].

Color features are rarely used in person detection because of the variability induced by clothing. But, color shows local similarity even over clothing. **Color Self Similarity (CSS)** features, proposed by Walk *et al.* [Walk 2010], encode similarities in different sub-regions. The features are computed first by subdividing the candidate window into non-overlapping blocks of 8x8 pixels and then computing a $3 \times 3 \times 3$ color histogram within each block (with interpolation). For each block, similarities are computed by intersecting individual block histograms. In [Walk 2010], all histogram intersections values are concatenated to define a single high dimensional feature vector.

Other features that have been used for people detection include: **Covariance features** [Tuzel 2008], which are an 8×8 covariance matrix computed in a rectangular region based on 8 extracted values, pixel position, absolute values of first and second derivatives (along

horizontal and vertical directions), gradient magnitude, and gradient orientation; **Shapelet features** [Sabzmeydani 2007], which are mid-level features learned from low level smoothed intensity gradients in a specified window; **Edgelet features** [Wu 2005], which are hard coded and pre-defined patterns of edges in different locations within the detection candidate window; and **Shape Context features** [Belongie 2002], which computed at a point express the configuration of the entire shape relative to the point using gradient magnitude and orientation pooled in a log-polar histogram.

Thus far the discussion has focused on features computed using only a single image frame. It is also possible to extract features considering consecutive image frames (incorporating temporal information). In the literature, two prominent features with this notion are the motion features extracted with **rectangular filters** [Viola 2005] and **Histogram of Flow (HOF)** of Dalal *et al.* [Dalal 2006b]. Viola *et al.* compute motion features based on motion images which are determined by differencing shifted version of the previous frame (shifted in four directions) with the current one. The actual feature values are extracted using rectangular differencing filters, placed at a precise position within the detection window, which compute differences amongst the different motion images or solely on a single one. Though simple, these features have shown to improve prior works based on appearance information only. On the other hand, HOF computation is rather involved. The authors first compute optical flow. Based on the optical flow, they proposed primarily two important variants: Motion Boundary Histograms (MBH) which are computed in similar fashion as HOG using gradient information of the motion flow images (angular voting is based on spatial derivative displacement of the flow images); and Internal Motion Histograms (IMH) computed like HOG with the angular voting based on the direction of the flow difference vector. Even though they obtained comparable results with both variants, the IMH variant showed better complementarity when combined with classical HOG. These features are best used with a static camera as their performance degrades with moving camera; the degradation is less pronounced with HOF features (with a considerate camera motion) because of the spatial binning.

Looking at the trend in the literature, the gist in features used for people detection can be captured with two important terms: gradient and histogram. The most successful features consider image gradients with local pooling in the form of histograms. This is evident considering peoples' global silhouettes, illumination and contrast variations in imaging, and deformation in physical structure. Gradient computation captures the intensity transition around peoples' body boundaries which helps furnish important information about structure of people, and computing the intensity difference (during gradient computation) helps with illumination and contrast invariance. The pooling in the form of a histogram makes the features robust with respect to small shifts, ameliorating deformation tolerance. In general, these considerations tend to lead to complex features that require increased computation time entailing more focus on computation time related optimizations. This being said, the next natural question would be, how about combination of features? Indeed, using a combination of features have shown to improve detection further, for example, the top 4 current best detectors (in terms of detection performance) in the state-of-the-art use a mixture of heterogeneous features (figure 2.7) [Dollár 2012].

Heterogeneous features help capture complementary information useful to handle various detection challenges–the more complementary the features the better. Many works in the literature have attested this complementary nature. Geronimo *et al.* [Gerónimo 2007] showed this with Haar like features and EOH; Wang *et al.* [Wang 2009] with HOG and LBP; Wojek *et al.* [Wojek 2008] with Haar like features, HOG, and shape context features; Walk *et al.* [Walk 2010] with a concatenation of HOG, HOF, and CSS. Similar conclusions were also made by Schwartz *et al.* [Schwartz 2009] and Hussain and Triggs [Hussain 2010] using—HOG, color frequency, and co-occurrence features—and—HOG and LBP variant features—respectively. With

a per frame (no temporal information) holistic representation, the best result in the literature is also obtained using heterogeneous features called **Integral Channel Features** [Dollár 2012]. Integral Channel Features denote a layer of several image channels computed using a unique image transformation per channel. In their implementation, Dollár *et al.* used three distinct feature families: color image, gradient image, and gradient histograms. Each component of the features goes into a specific channel (3 channels for color in LUV space, 1 channel for gradient magnitude, and 6 orientation binning channels for each gradient orientation) and then specific features such as local sums, histograms, and Haar like features are computed per each channel efficiently using integral images.

Given heterogeneous pool of features, different ways can be used to build the final composite feature. Four main trends are observed in the literature: (1) Direct concatenation [Walk 2010, Wojek 2008] in which the different features are concatenated to make one high dimensional feature vector; (2) Feature selection where a subset of the efficient features are selected using one of the boosting techniques [Schapire 2003]; (3) Coarse-to-fine hierarchical arrangement [Mogelmose 2012, Pan 2013] where a cascade is constructed using cheap features at the initial stages and using complex features at later stages; and (4) Multiobjective optimization with respect to computation time and detection [Jourdheuil 2012, Wu 2008]. The downsides of direct concatenation are, first the increased computation cost owing to the complex feature constructed, and second classification in the merged space that could be disadvantageous as different features might possibly be best dealt with different classifiers—they could for example lie in linear or non-linear spaces which may require different classification techniques. On the other hand, feature selection with a boosted classifier suffers with respect to computation time as classical boosting classifiers select features solely based on detection performance, favoring complex features, even though a combination of cheap features might achieve the same performance. The coarse-to-fine hierarchy is quite advantageous and tries to find a balance between detection performance and speed. The concern is how to decide which features to use at the different stages systematically? For example, both [Mogelmose 2012, Pan 2013] adopt a heuristic based rule and use homogeneous family of features they deemed cheap at the initial stages, and homogeneous complex features at the latter. Finally, the multiobjective optimization approach is the most appealing if the optimization is done with respect to feature computation time and detection performance. In chapter 4, a noble framework is proposed that merges the notions of the coarse-to-fine hierarchical arrangement with a multiobjective optimization. The framework selects features that achieve a stipulated detection performance and have the minimum combined computation time; this systematically leads to a detector with a coarse-to-fine hierarchical arrangement.

2.2.2.5 Classification

The classification stage is responsible for labeling each candidate window generated and described in accordance with the abstraction adopted as either a person or not. This block can either output a binary label (person or non-person) or a continuous valued score that reflects its confidence, and can further be thresholded to provide a binary label. These classifiers are mostly trained with a discriminative learning algorithm given positive and negative example instances. As stated previously, discriminative learning algorithms in addition to robust image representation are the key reasons to recent advances in people detection. The most frequently used discriminative classifiers for people detection are variants of Support Vector Machines (SVM) and Boosted classifiers. On few occasions Fisher's Linear Discriminant Analysis (LDA), *e.g.*, [Paisitkriangkrai 2008] and Artificial Neural Networks, *e.g.*, [Szarvas 2005, Zhao 1999] have also been used; recently, Random Forest classifiers are also gaining attention [Tang 2012].

SVMs are statistical supervised learning algorithm used for classification and regression introduced by Vladmir Vapnik [Vapnik 1995]. With SVM the classification boundary is the one that maximizes the margin between the different classes of the training data. It is considered a good classifier candidate because of its high generalization performance without the need to add a priori knowledge, even when the dimension of the input space is very high [Vapnik 1995]. As a result, it has been extensively used for people detection: linear SVMs with holistic abstraction, *e.g.*, [Dalal 2005, Wang 2009]; non-linear SVMs, *e.g.*, [Maji 2008]; latent SVM with parts-based abstraction [Felzenszwalb 2010b].

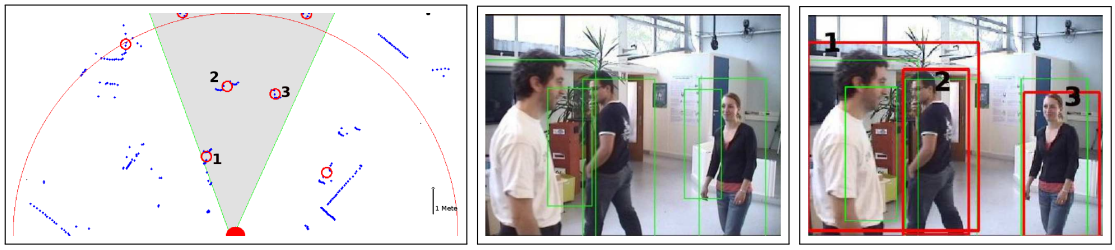
Boosted classifiers, also called ensemble classifiers, construct a strong classifier by combining weak classifiers where each consecutive weak classifier focuses on previously misclassified problems. They are quite appealing as they perform feature selection automatically. Different variants have been proposed and applied for people detection in the literature. Pertinent variants include: Discrete AdaBoost, *e.g.*, [Viola 2004], Real AdaBoost, *e.g.*, [Gerónimo 2010a], LogitBoost, *e.g.*, [Tuzel 2008]. One appealing characteristics of boosting is the liberty to choose the weak classifier; theoretically the weak classifiers need to do better than chance for the algorithm to work. In practice, the choice with the weak classifier has varied from a simple decision stump to an SVM with improved result guaranteed in all cases compared to individual ones. In the literature, a boosted classifier is almost always constructed in an attentional cascade architecture, also called rejection cascade, that has the form of a degenerate tree [Viola 2004]. Each node of the cascade is trained with a subset of the training sample whereby initial nodes will tackle simpler problems and latter nodes will face difficult ones. This way only stronger hypotheses will be evaluated by all nodes increasing speed drastically. Boosted cascade is the prominently adopted framework whenever detection speed is of concern, *e.g.*, [Viola 2004, Dollár 2009, Felzenszwalb 2010a].

2.2.3 Multi-modal Approaches

Multi-modal approaches try to build a better detector by utilizing multiple detection modes with the help of various sensors. The motivation is to use different detectors with different modalities to build a better detector as no single detector system is perfect. This approach is very common in Robotic applications as most robotic platforms are equipped with multiple sensors. By combining information from different sensors, the shortcomings of one sensor can be compensated by another one leading to better detection performance. The key question here is: how to actually combine the (heterogeneous) data from the different sensors?

A very straight forward technique with a sensor that is computational cheap to process –and yet furnishes unreliable detections–and a sensor that provides rich information leading to precise detections at the cost of expensive computational resources is a sequential chain. The idea is to minimize the number of hypothesis that need to be examined with the precise sensor by first utilizing the information from the cheap sensor. A very good set of exemplary sensors used in the literature are a visual camera (classical or omnidirectional) and a laser range finder [Mekonnen 2011]. In our work Mekonnen *et al.* [Mekonnen 2011] hypothesis based on a laser range finder data is used to constrain the search in the visual data leading to an improved detection performance and speed, figure 2.6 shows some illustrative images. The downside is detections that are missed in the initial hypothesis set would be completely misdetected by the combined system.

An alternative approach processes the information from the different sensors independently and fuses the detections by applying decision rules that reinforce common interpretation and resolve differences. Contrary to ad-hoc or heuristics based rules, fusion decision rules that rely on probabilistic fusion techniques are sound and theoretically appealing. Frequently used probabilistic fusion techniques include Kalman Filter variants [Lefebvre 2004], Particle



(a) (b) (c)

Figure 2.6: Illustration of a sequential multi-modal people detector. (a) shows raw laser scans with people detection hypotheses (circled in red). In (b), the hypotheses are projected on to the image, shown in green rectangles (the field of view of the camera corresponds to the shaded region in (a)). Finally, (c) shows confirmed detection (in red) determined by running a computationally expensive visual people detector. This avoids brute force scanning of the image for persons leading to a drastic speed-up.

Filter variants [Chen 2003], Covariance Intersection [Chen 2002], and Probabilistic Anchoring [Elfring 2013]. These techniques lead to a tractable formulation that cater spatio-temporal detection information of targets–commonly referred to as target tracks. Target tracking is addressed extensively in the second part of this thesis. It is also possible to fuse different detections in a probabilistic framework to obtain a robust detector without any temporal inference. For example, Zivkovic and Kröse [Zivkovic 2007] used a 2D Laser Range Finder and an omni-directional visual camera mounted on a robot for people detection. The authors combined leg detections from a laser range finder and detected human parts–full body, upper body, lower body–from the image in a parts based probabilistic model. The final detection decision is made using a maximum likelihood estimate.

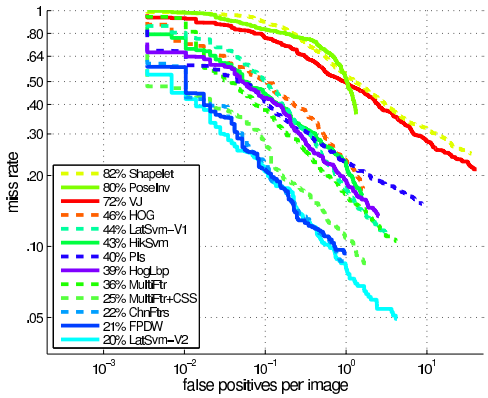
2.2.4 Discussions

In this section the different trends and modes for automated people detection in the literature have been presented. The section started with a discussion of the different sensors that have been used for people detection. Generally speaking most of the active sensors tend to provide precise and easy to deal with information. However, the most informative as well as cost effective sensor is a visual camera, which justifies the overwhelming attention it has gotten. In addition, contrary to robotic system, a visual camera is usually the sole source of information in video surveillance and image indexing applications. In robotic contexts, a multi-modal approach, where multiple detection modes from heterogeneous sensors are considered, is mostly privileged (considered in detail in part II of this thesis).

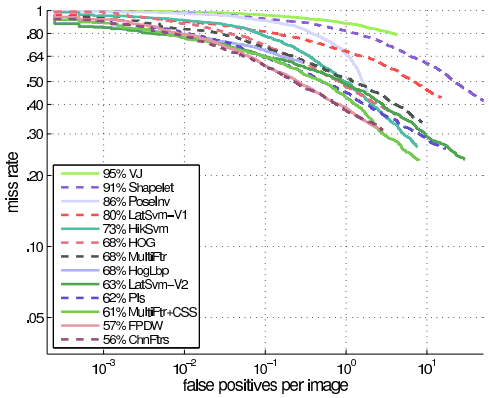
In visual people detection, the detection pipeline has three main stages (figure 2.1). The candidate window generation step, the image representation step, and classification step. The pipeline makes use of an underlying person abstraction. Implicit abstractions make assumptions either on the camera configuration (*e.g.*, static camera) or the scene (*e.g.*, only people move). On the other hand, explicit abstractions refrain from imposing these constraints and describe people either as a holistic object or an object made up of parts with connections. The sliding window approach as a candidate generation step is the most widely used as it does not miss any hypothesis and does not rely on any prior camera or scene knowledge. The most important features used in the literature have been presented in section 2.2.2.4. The two most important class of classifiers are SVM and Boosting variants. Boosting variants are mostly used to build a cascaded detector systematically arranging cheaper features (either simple or fewer in number)

in the initial stages and complex ones at the later stages drastically improving detection speed.

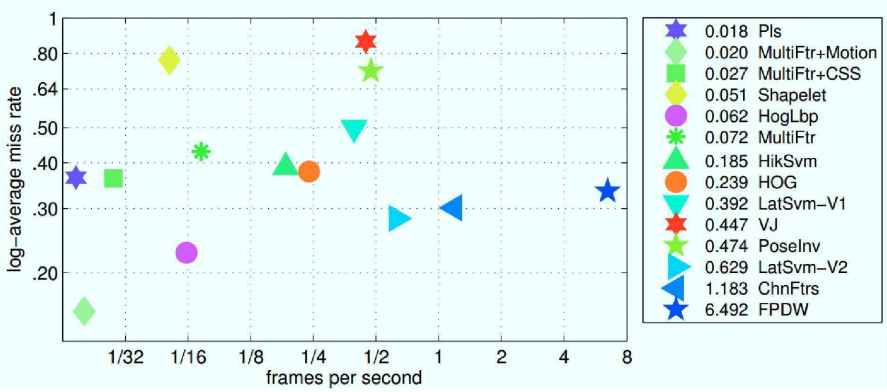
With this in mind figure 2.7 shows the performance of the prevalent vision based people detection methods proposed in the literature. Figures 2.7a and 2.7b show the detection performance on the INRIA person dataset (section 2.3.2) and Caltech pedestrian dataset (section 2.3.3) respectively using the full image evaluation scheme discussed in section 2.4.3. The computation time of each of these methods is also reproduced in figure 2.7c, from [Dollár 2012]; though the values reported are specific to the actual machine used for evaluation, they help provide a clear idea about the relative differences.



(a) Evaluation on INRIA test dataset.



(b) Evaluation on Caltech test dataset (reasonable category).



(c) Log-average miss rate versus the runtime of each detector on 640 × 480 images for pedestrians over 100 pixels, *i.e.*, *near scale*, from the Caltech Pedestrian Dataset (taken from [Dollár 2012]).

Figure 2.7: Performance of the state-of-the-art people detectors (produced using the toolbox from [Dollár 2012]). Please refer to section 2.4 for explanation of the different evaluation metrics.

The following methods, all which rely on a sliding window candidate generation mode, are represented:

- **VJ**: Refers to the work of Viola and Jones [Viola 2004] based detector which employs Haar like features with AdaBoost and a holistic person abstraction.
- **Shapelet**: [Sabzmeydani 2007], which uses mid-level features learned from low level

smoothed intensity gradients in a specified window (called shapelet features) combined with a boosting discriminative classifier to form an overall detector.

- **PoseInv**: A parts-based detector that uses HOG features extracted along people's shape outline [Lin 2008].
- **HOG**: Dalal and Triggs [Dalal 2005] people detector which introduced and used HOG features with an SVM classifier in a holistic paradigm.
- **LatSvm-V1**: The most successful parts-based approach of Felzenszwalb *et al.* [Felzenszwalb 2010b] using HOG features and latent-SVM classifier. Its variant that uses a cascaded configuration with a model trained on the INRIA person dataset is shown as **LatSvm-V2** [Felzenszwalb 2010a].
- **HikSvm**: A holistic abstraction with multilevel HOG like features and non-linear SVM with an approximated histogram intersection kernel [Maji 2008].
- **Pls**: A detector that uses edge, texture, color features with a dimensionality reduction step using partial least squares and SVM as a classifier [Schwartz 2009].
- **HogLbp**: [Wang 2009], combined HOG and LBP features, computed and concatenated over a candidate window, with a linear SVM.
- **MultiFtr**: A holistic approach with a high dimensional feature composed of Haar like, shapelets, shape context, and HOG [Wojek 2008]. These feature sets were later extended to incorporate CSS features, **MultiFtr+CSS**, and motion features with HOF feature,**MultiFtr+Motion**, [Wojek 2008].
- **ChnFtrs**: Corresponds to works of [Dollár 2009] that use Integral Channel Features with Boosting classifier. **FPDW** uses the same underlying features used by **ChnFtrs**. But, it optimizes the detection process by approximating the features over scale space resulting in a fast multi-scale detector.

From the above brief description of each detector and reflected performance shown in figures 2.7 and 2.7b the following important observations can be made concerning detection performance: (1) Approaches that use heterogeneous pool of features excel in terms of detection performance; (2) Using the same kind of features, parts-based approaches result in improved detection (*e.g.*, **HOG** vs **LastSvm**); and (3) With the exception of **VJ** the rest of the methods rely on some variant of gradient orientation and magnitude based features. These points clearly indicated the way to go to achieve a state-of-the-art detector from the detection performance perspective. If we look at the computation time aspect, it can be observed from figure 2.7c that detectors that use a cascade configuration or computationally cheap features generally result in faster detectors. Considering direct concatenation of heterogeneous features with a one shot classifier (like SVM) does not benefit detection speed. A combination of cascade configuration with cheap to compute features actually leads to more faster detectors.

Based on these insights we propose two people detectors that use homogeneous features (HOG) in chapter 3 and heterogeneous pool of features in chapter 4. An approach based on HOG is investigated initially as it is the most discriminant feature. Then, considerations are given to heterogeneous pool of features since only heterogeneous features based approaches have managed to supersede HOG in general. In both cases a cascade configuration is privileged. Contrary to classical approaches, we propose a novel feature selection paradigm based on discrete optimization that selects a subset of features that fulfill the stipulated detection performance and

at the same time result in the lowest possible combined computation time. A thorough evaluation of the proposed detectors is also presented based on a proprietary dataset (section 2.3.1) and public datasets (sections 2.3.2 and 2.3.3) using the the evaluation metrics presented in section 2.4. To further improve perception of people in an environment, multi-modal approaches that fuse detections from multiple sensors are presented and discussed in part II of this thesis.

2.3 Visual Datasets

2.3.1 Ladybug Dataset

The Ladybug dataset is a custom compiled dataset using images acquired with the *Ladybug2* camera in our robotic laboratory. This dataset features images of people acquired in a very cluttered indoor environment. The *Ladybug2* (figure 2.8a), manufactured by Point Grey Inc [Point Grey Inc. 2012], is a polydioptric camera that provides real omnidirectional view without pronounced geometric, resolution, and/or illumination artifacts. It is a spherical omnidirectional camera system that has six cameras mounted in such a way to view more than 75% of a full sphere. Each camera has a maximum resolution of 1024×768 pixels resulting in a 3500×1750 pixels stitched high resolution panoramic image (figure 2.8b). This high resolution image entails high computational resources for processing. The camera system has an IEEE-1394b (FireWire 800) interface that allows streaming at 30 fps with the drivers provided by the manufacturer [Point Grey Inc. 2012].

(a) (b)

Figure 2.8: *Ladybug2* camera and a corresponding stitched image.

The omnidirectional image, produced by stitching the six images, nevertheless introduces a geometrical distortion when the stitched spherical image is unrolled into a planar image. Particularly, the aspect ratio of people that are close to the camera changes. Hence, this dataset is prepared for training a detector tuned for this camera/image set.

This dataset consists of two distinct sets. The first one, referred as the training set, consists of 1990 positive samples (original and mirrored version) annotated by hand and scaled to a 128×64 pixels window. It also contains 58 person free full resolution images acquired from our robotic and other rooms in the laboratory. A total of $488,992$ negative windows of 64×128 pixels are uniformly sampled from these person free images. Figure 2.9 and 2.10 show sample positive and negative windows taken from this set respectively.

The second set used for testing purposes—hence, called the test set—contains $1,000$ original and mirrored manually cropped positive samples of 128×64 pixels and 41 person free images, out of which $319,653$ negative windows are uniformly sampled. Similar illustrations for the positive and negative samples are shown in figure 2.11 and 2.12 respectively.

Figure 2.9: Sample positive images taken from the Ladybug training dataset.

Figure 2.10: Sample negative images taken from the Ladybug training dataset.

Figure 2.11: Demonstrative positive images taken from the Ladybug test dataset.

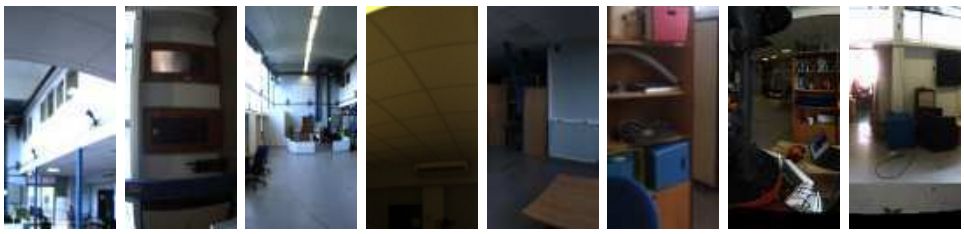


Figure 2.12: Demonstrative negative images taken from the Ladybug test dataset.

2.3.2 INRIA Person Dataset

The INRIA person dataset [1], introduced by Dalal and Triggs [Dalal 2005], is the most important and widely used public dataset for benchmarking purposes in people/pedestrian detection works. The dataset is divided in two formats, a format which contains original images (full images with people in natural scenes) with corresponding annotations and a second format which contains cropped positive images and people free negative images. Since the second format has been extensively used in the literature, including the original published work on it [Dalal 2005], it is

[1] The INRIA person dataset can be downloaded from http://pascal.inrialpes.fr/data/human/

described in detail here and is used in the rest of this work.

(a) Sample positive training images.

(b) Sample negative training images.

(c) Sample positive test images.

(d) Sample negative test images.

Figure 2.13: Illustrative samples from the INRIA person dataset.

This format consists of both training and testing sets. The training set features 2416 cropped positive instances (originals and mirrored versions) and 1218 images free of persons. The cropped instances have a resolution of 160×96. But, the actual size of pedestrians bounding is 128×64. The extra padding is provided to minimize the border effect. In this work, a total of 2.55×10^6 negative cropped instances are generated from these person free images. The test set contains 1132 positive instances and 453 person free images. Similarly, 2×10^6 cropped negative windows are uniformly sampled from the person free images during testing. Sample cropped images from this dataset are shown in figure 2.13.

The cropped positive samples in both the training and test set are obtained from manually annotated real life images. Sometimes, it is interesting to report detector performance on a per image basis rather than using the cropped sets, especially with regards to performance on the test set. For this purposes, this dataset also contains the original 288 images out of which the cropped positive test samples are collated.

2.3.3 Caltech Pedestrian Dataset

The total Caltech Pedestrian Dataset [Dollár 2012] consists of about $250,000$ image frames of size 640×480 acquired from a vehicle driving through regular traffic in a city. The annotation totally contain $350,000$ bounding boxes consisting of $2,300$ unique pedestrians. The dataset is divided into 11 different sets (S0-S10), six training sets (S0-S5) and five testing sets (S6-S10). Figure 2.14 shows sample annotated frames. Dollár *et al.* [Dollár 2012] recommend training a detector using either an external dataset or the Caltech training set (S0-S5), and testing the finalized detector on the Caltech testing set (S6-S10).

Evaluation results on this dataset are reported using the full image evaluation metrics, section 2.4.3. Due to the overwhelming size of the dataset and computation demand of detectors, test evaluations are performed on each 30^{th} frame starting from frame 30. The annotation also contains pedestrian appearance characteristics that enables segregation of the evaluation into the following groups: *near scale*, pedestrians over 80 pixels tall; *medium scale*, 30 to 80 pixels tall pedestrians; *far scale*, under 30 pixels tall; *no occlusion*, unoccluded pedestrians over 50 pixels; *partial occlusion*, pedestrians with partial occlusion (1-35% area occluded); *heavy occlusion*,

Figure 2.14: Illustrative image frames taken from the Caltech pedestrian dataset with overlaid people bounding box annotations.

pedestrians with 35-80% occluded surface; and *reasonable*, 50 pixels or taller pedestrians with no or partial occlusion. It is also possible to do the evaluation on all categories to get *overall* performance statistics.

2.4 Evaluation Metrics

In the literature, various metrics have been proposed and used to evaluate different person detectors. The metrics are used into two differing evaluation schemes: (1) a Per Window (PW) evaluation scheme, and a Full Image (FI) evaluation scheme. As the name clearly suggests, the PM method determines performance based on cropped positive and negative image windows. This approach isolates classifier performance from overall detection system, thus, making it ideal for characterizing classifier performance [Dollár 2012]. On the other hand, the FI methodology relies on detection outputs provided on an input image in the form of bounding boxes. The scheme makes the assumption that the detector under evaluation performs multi-scale detection and appropriate non-maximal suppressions. Hence, this scheme is most suitable for evaluation complete detection systems.

The work presented in the forthcoming chapters revolves around features, classifiers, and complete detectors. Hence, both the PW and FI methodology are used to report results. With the PW scheme, we present two mostly used metrics: the Detector Error Tradeoff (DET) curves [Dalal 2005] and Precision-Recall curves used the Pascal Visual Object Classification (VOC) challenge [Everingham 2010].

2.4.1 PW Evaluation: Detector Error Tradeoff (DET)

This evaluation metric depicts Detection Error Tradeoff (DET) curve with Miss Rate (MR) versus False Positives Per Window (FPPW) on a log-log scale [Dalal 2005]. To determine these values the True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) of the test set are determined by tracking detection success and failure on cropped labeled positive and negative windows. The plot is generated by operating the classifier at varying operating points (usually obtained by varying the classifier threshold) and computing the MR (equation 2.1), which is 1− True Positive Rate (TPR), and FPPW (equation 2.2) at each point.

$$MR = 1 - TPR = 1 - \frac{TP}{TP + FN} \tag{2.1}$$

$$FPPW = \frac{FP}{FP + TN} \tag{2.2}$$

This metric is principally used to report comparative results of the proposed detector with other detectors in the literature.

2.4.2 PW Evaluation: Precision-Recall

The Precision-Recall curve is also a well established and commonly used metrics in object detection/classification tasks; it has been extensively used in the Pascal Visual Object Classification (VOC) challenge [Everingham 2010]. The evaluation involves a Precision-Recall curve and a single scalar quantity called Average Precision (AP), which is basically the area under the Precision-Recall curve computed by taking the mean precision at a set of eleven equally spaced recall levels. The Precision and Recall, equations 2.3 and 2.4 respectively, are computed at different operating points of the classifier/detector using the PW scheme.

$$\text{Recall} = TPR = \frac{TP}{TP + FN} \tag{2.3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.4}$$

These metrics are quite useful when it is necessary to compare competing detectors with comparable performance using AP which summarizes overall performance in a single quantity.

2.4.3 Full Image Evaluation

The full image evaluation scheme is very useful to evaluate the performance a complete detector system. The idea is to give an annotated (with bounding boxes corresponding to people) image to a detector system and then evaluate its performance by counting TP, FP, and FN occurrences cross checked with the annotation. The most widely used evaluation in this category uses the Miss Rate vs False Positive Per Image (FPPI) plot. The FPPI is basically the FP occurrences averaged over the number of tested image frames. A detection is considered a TP only if the overlap between the detection bounding box (BB_d) and the ground truth bounding box (BB_g) is sufficient (above a pre-determined threshold, equation 2.5).

$$Overlap = \frac{area(\text{BB}_d \cap \text{BB}_g)}{area(\text{BB}_d \cup \text{BB}_g)} \tag{2.5}$$

In [Dollár 2012] and Pascal VOC [Everingham 2010], a TP is counted only if $Overlap > 0.5$, otherwise it is counted as a FP. In recent years, this metric is the prominently used and in this work it is especially used when comparing results with the state-of-the-art. To summarize the performance of a detector, the *log-average miss rate*, computed by averaging the miss rates at nine evenly spaced (in log-space) FPPIs, is used.

2.5 Summary

In this chapter the different trends, modes, and considerations in automated people detection have been presented. The chapter starts out with the challenges in automated people detection and then proceeds with a presentation of different sensors that have been utilized for detection. Part of the chapter has been dedicated for a thorough presentation on visual camera based approaches (in line with its share in the literature pool and subsequent consideration). Different evaluation metrics used as well as public and proprietary datasets have also been presented. The material set forth in this chapter will help put the developments and contributions in subsequent chapters in perspective.

Chapter 3

Optimized HOG based Person Detector

Contents

3.1 Introduction **39**
3.2 Framework **40**
3.3 HOG-based Feature Set **41**
3.4 Weak Learners **43**
3.4.1 Fisher's Linear Discriminant Analysis 44
3.4.2 Support Vector Machines 45
3.5 Pareto-Front Analysis **45**
3.6 Feature Selection via Binary Integer Programming **46**
3.7 Discrete AdaBoost and Cascade Construction **48**
3.8 Experiments and Results **49**
3.8.1 Implementation Details and Validation 50
3.8.2 Results 53
3.8.3 GPU Implementation 58
3.9 Discussions **58**
3.10 Conclusions **62**

3.1 Introduction

As discussed in the previous chapter, automated people detection finds application in many areas including, but not limited to, human-robot interaction, surveillance, pedestrian protection

systems (in the automotive industry), and automated image and video indexing. At the same time, it is by far challenging due to physical variation of persons, articulated poses, highly variable appearances because of clothing, view point variation, occlusion, background clutter, *etc.* In the previous chapter, the trends, different modes, and considerations related to automated people detection have been discussed in detail. The content mainly focused on vision based approaches in line with their importance in the literature. Recall that when dealing with vision based people detection, the two factors that should be taken into consideration are detection performance and computation time. A majority of the approaches in the literature make significant contributions in either one of these factors but not both, though a handful of the approaches have managed to do so in both factors. In this chapter we will present a detection framework that tries to take both constraints into consideration.

We present a person detector with a cascade configuration that uses the most discriminant HOG feature tweaked to be suited for feature selection. To address both detection performance and computation time constraints we propose a novel discrete optimization technique based on Binary Integer Programming (BIP). Consequently, this chapter makes three main contributions: (1) a sound mathematical formulation based on BIP for feature selection taking both computation time and detection performance into consideration, (2) implementation details of a detector based on the above formulation, and (3) a thorough and comparative evaluation of the proposed detector with the prominent approaches in the literature. It starts with a presentation of the framework in section 3.2, followed by a description of each component of the framework in sections 3.3 to 3.7. Experiments and results are presented in section 3.8 and finally the chapter finishes with discussions and conclusions in sections 3.9 and 3.10 respectively.

3.2 Framework

Evidently, the framework adopted to address the aforementioned objectives needs to be concerned by detector detection performance and its associated computation time. The most famous detector configuration suitable for these requirements is the attentional cascade detector configuration pioneered by Viola and Jone [Viola 2004]. This configuration builds a cascade made of nodes resembling a degenerate tree. Each node rejects negative candidate windows and passes along potential positive windows onto the next stage for more scrutinized verification. Figure 3.1 illustrates this configuration made up of K nodes. Given a candidate window, it is passed, with a label T for true, along the cascade only if it fulfills the test encountered at each node, otherwise it gets rejected (labeled as F for false). A window is considered to be positive only if it makes it to the end of the cascade, *i.e.*, only those classified as true detection by all nodes are considered as true targets. This leads to an efficient structure that uses simple classifiers at the beginning of the cascade, which rejects a majority of the negative samples, and complex classifiers as one progresses along the cascade speeding up detection drastically. This structure has gained wide acceptance and has even been applied in recent part-based approaches [Felzenszwalb 2010b].

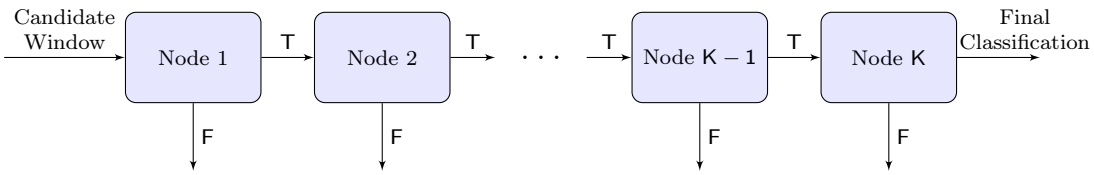


Figure 3.1: An attentional detector cascade configuration.

Figure 3.2 shows the proposed framework to train a classifier that will be used on each node

of the cascade. The first important point to notice is, instead of using the classical HOG feature vector concatenated from the entire candidate window, we propose to concatenate a varying number of HOG blocks to obtain a pool of HOG features, $\mathcal{F}$, as described in detail in section 3.3. Then for each feature in the feature set, a classifier is learned using the input training set $\{(x_i, y_i)\}_{i \in N}$ composed of n training samples ($N = \{1, 2, ...,n\}$) where x_i denotes the i^{th} training sample with label $y_i \in \{-1, +1\}$. The weak classifiers are the distinct classifiers trained on the given training samples per single feature (a single classifier is trained per each feature). Hence, a feature and its weak classifier are coupled and either reference can be used to refer one or the other. The different weak classifiers considered are detailed in section 3.4.

At this point, each feature (along with its trained weak classifier) can be characterized by three parameters: True Positive Rate (TPR) and False Positive Rate (FPR) determined on a validation set, and its computation time τ. Given this information, a two step feature selection procedure is applied to select a subset of features that fulfill a stipulated detection performance and result in the smallest cumulative computation time. First, a Pareto-Front analysis [Chong 2008] is carried out to retain dominant features with respect to TPR, FPR, and τ (explained in detail in section 3.5). This step is employed to reduce the total number of features to a tractable size, $\mathcal{F} \rightarrow \tilde{\mathcal{F}}$, for the second step—feature selection via discrete optimization. The discrete optimization step for feature selection is realized using Binary Integer Programming (BIP). BIP is a special case of integer programming where decision variables are required to be 0 or 1 (rather than arbitrary integers). The BIP results in a few set of selected features, $\hat{\mathcal{F}}$, that fulfill the detection requirement with the smallest cumulative computation time (section 3.6).

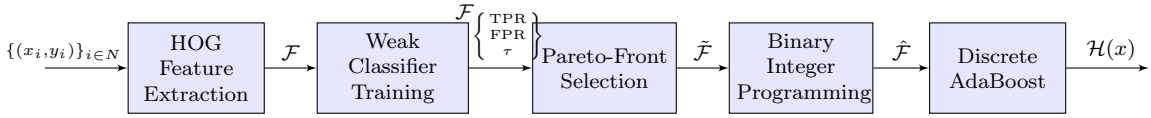


Figure 3.2: Feature selection and classifier learning framework used at each node of a cascade.

Finally, Discrete AdaBoost takes the features selected by the BIP, $\hat{\mathcal{F}}$, and builds a strong classifier, $\mathcal{H}(x)$, by weighting and combining them as described in section 3.7. This classifier learning framework is applied at each node of the cascade until all training samples are completely exhausted resulting in the final cascaded classifier.

3.3 HOG-based Feature Set

As it has been mentioned in the previous chapter, no other single feature has been able to supersede HOG feature [Dollár 2012]. Hence, naturally, we have resorted to use it. As illustrated in table 3.1, HOG feature computation starts by a gamma normalization followed by first order gradient computation. Then, a gradient magnitude histogram over gradient orientation bins is constructed in each atomic regions called a *cell*. Histograms of neighboring cells are grouped into a single *block*, cross-normalized and concatenated to give a feature vector per block. The final extracted feature within a given detection window is the concatenation of the vectors from each feature block. In its most widely used configuration, 9 contrast insensitive orientation histogram bins with linear interpolation are used. Each block encompasses 2×2 cells each of which are made up of 8×8 pixels. The horizontal and vertical block stride is set to 8 pixels which results in each cell contributing to four different blocks (this results in an 8 pixels overlap between consecutive blocks). With these configurations, the dimension of the final concatenated HOG

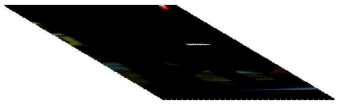
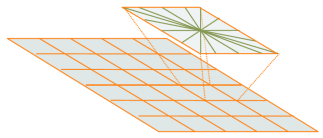
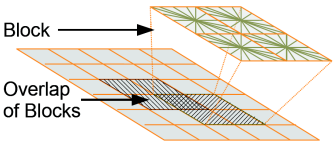
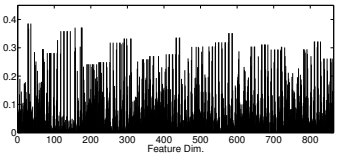
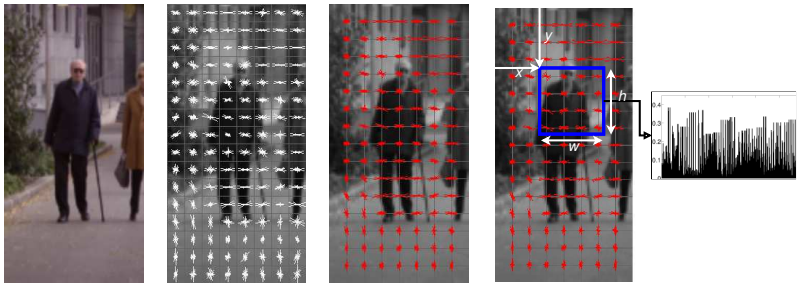
Illustration	Description
	Input candidate window of 128×64 pixels in size.
	To reduce the influence of illumination effects, a gamma compression is applied by computing the square root of each color channel.
	First order image gradient computation. The gradient is computed for each pixel and for color images the dominant gradient channel is retained.
	Divide the window into non-overlapping spatial regions called *cells*. In each *cell*, construct a 1-D histogram by accumulating gradient magnitude weights over gradient orientation bins.
	Cells are grouped into overlapping ensembles called *blocks*. Because of the overallaping, a cell is shared between several blocks. Within each block, all cell histograms are normalized by the local histogram "energy" of the block. The cell histograms of a block concatenated gives the *block histogram*.
	Finally, all block histograms computed within the candidate window are concatenated to give the HOG feature vector.

Table 3.1: Illustration of HOG features computation. (Illustrations taken from [Dalal 2006a].)

feature vector for a 128×64 candidate window becomes 3780. Evidently, the main culprits for the high computation time are extracting this high dimensional feature vector and applying the associated classifier weights of equal length on all incoming candidate windows.

In this proposed approach, we use the original HOG features proposed by Dalal and Triggs [Dalal 2005] along with their widely preferred/used computation, *i.e.*, a cell size of 8×8 pixels, a feature block size of 2×2 cells and an 8 pixel horizontal and vertical stride. But, instead of using the entire descriptor as a single feature, we generate a pool of features by concatenating only a subset of the block histograms. The main steps are illustrated in figure 3.3. Given a candidate window (figure 3.3a), cell histograms are computed according to table 3.1 and are shown in figure 3.3b. Similarly, block histograms are computed as described in table 3.1, figure 3.3c. Then,



(a) (b) (c) (d)

Figure 3.3: Illustration of the HOG feature pool set generation.

differing from the original proposition, a single feature is described by concatenating all block histograms inside a region parameterized by a starting location (x,y), width (w), and height (h) as shown in figure 3.3d. Finally, the entire features in the feature pool are generated by varying x, y, w, and h for all possible positive values in the given candidate window. If we consider Ω to be an operator that concatenates all block histograms in a given region and that there are 15×7 block histograms in the specified candidate window, the feature set, $\mathcal{F}$, can be expressed using equation 3.1. This leads to a total of 3360 features with dimensions ranging from 36 (smallest) to $7 \times 15 \times 36 = 3780$ (highest and equivalent to the feature vector determined by concatenating all block histograms).

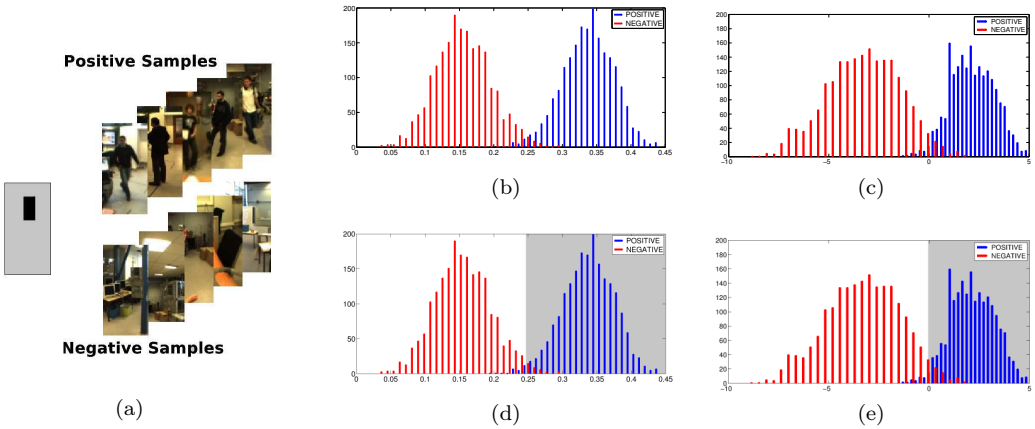
$$\mathcal{F} = \{\Omega(x, y, w, h) : 0 \leq x < 7; 0 \leq y < 15; 1 \leq w \leq (7 - x); 1 \leq h \leq (15 - y)\} \tag{3.1}$$

In summary, in the works of Dalal and Triggs, all resulting feature blocks extracted from the 128×64 image window are concatenated, giving a single high dimensional vector—with exactly $7 \times 15 \times 36$ dimensions—as a final feature. Whereas, in our case, we end up with a total of 3360 variable dimension pool of features.

Computation Time: The features in our feature pool are of varying dimensions. Incidentally, the associated time taken to extract them varies. Since the smallest building unit is a single HOG feature block, determining the computation time of each feature obtained using the above defined Ω operator is straight forward. Each feature obtained using Ω contains an integral multiple of individual HOG blocks. If it takes τ milliseconds to compute the feature vector of a single block, then it takes $n.\tau$ milliseconds for a feature made up of n blocks using the Ω operator. With this, the computations time for the different features in the pool varies from the smallest, τ, to the highest, $105.\tau$ milliseconds.

3.4 Weak Learners

All features in the presented feature set, *i.e.*, $\mathcal{F}$, are multi-dimensional vectors with dimensions ranging from 36 to 3780. In the literature, the most prominent classifiers used with multi-dimensional feature vectors are variants of SVMs. In fewer occasions Linear Discriminant Analysis (LDA) has also been investigated, *e.g.*, [Paisitkriangkrai 2008]. Consequently, in this section, weak learners based on LDA and SVM are discussed. In consecutive references, a weak learner trained using a feature indexed by j from $\mathcal{F}$ is denoted as $\mathfrak{h}_j$.



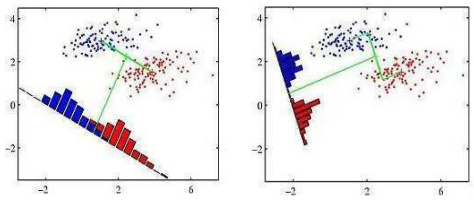
(a) (b) (c) (d) (e)

Figure 3.5: Weak learners classification illustration. (a) A few samples of the positive and negative images from which HOG features with the small black rectangular spatial support are extracted. (b) A histogram of projected scalar values using the projective vector determined via Fisher LDA. (c) A similar histogram determined using the trained linear SVM hyperplane. (d) Shaded region showing space classified as positive determined after training a 1-D classifier on top of the LDA projected values (using a single threshold, *i.e.*, DT of depth 1). (e) Here, the shaded region shows the region classified as positive which is greater than zero (as per SVM classification rule).

3.4.1 Fisher's Linear Discriminant Analysis

Fisher's Linear Discriminant Analysis (LDA) [Fisher 1936] belongs to a set of techniques that seek to reduce the dimensionality of a categorized data while preserving as much of the category discriminatory information as possible. Given a labeled input data $\{(x_i, y_i)\}_{i \in N}$ where $x_i \in \Re^d$, $y_i \in \{-1, +1\}$, and d is dimension of the feature vector, Fisher's LDA tries to find a projection vector $\mathbf{w}$ that upon data projection results in the maximum separability of the projected scalar values, the vector shown in figure 3.4b rather than figure 3.4a.

It is looking for a projection vector where samples from the same class are projected very close to each other and, at the same time, the projected means are as far apart as possible. Specifically, Fisher's LDA determines a projection vector $\mathbf{w}$ that maximizes $J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$, where S_B is the "between classes scatter matrix" and S_W is the "within classes scatter matrix". But, for classification tasks, we should not stop here. Once the best $\mathbf{w}$ is determined and all high dimensional training data is projected to a scalar value, the problem reduces to a one dimensional (1-D) classification task. For the 1-D classification, we then apply a Decision Tree (DT) to classify the data. The DT is equivalent to having multiple thresholds depending on its depth (a depth of 1 corresponds to a single threshold whereas more depth means more interleaved classification intervals). Figure 3.5 shows a sample classification using Fisher LDA on a single arbitrarily chosen HOG feature.



(a) (b)

Figure 3.4: Two projection vectors, LDA results in the second vector, (b), as it maximizes class separation.

3.4.2 Support Vector Machines

Support Vector Machines are statistical supervised learning methods used for classification and regression. SVMs are the dominantly used classifiers in detection applications that involve high dimensional training data. This is so because of their high generalization ability without the need to add *a priori* knowledge even with very high dimensional input space [Vapnik 1995].

Similar to LDA, given a perfectly separable labeled input data $\{(x_i, y_i)\}_{i \in N}$ where $x_i \in \Re^d$, $y_i \in \{-1, +1\}$, and d is dimension of the feature vector, SVM formulation aims to determine a hyperplane, $\mathbf{w} \cdot x_i + b = 0$, that separates the datasets into their respective classes, *i.e.*, $y_i \cdot (\mathbf{w} \cdot x_i + b) > 0$ for all $i \in N$, while maximizing the margin between the two classes. The solution that satisfies these requirements is a hyperplane determined by $\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i y_i x_i$ and the class label for a given example x becomes $sign\,(\mathbf{w}_o x + b)$. (The α_i terms will be zero for most of the trainig dataset x_i except for a few which are called *Support Vectors*.) As can be observed, the classifier just presented is based on a linear hyperplane in the original feature vector space and hence is called Linear SVM. The formulation also extends for a non-separable class by incorporating a penalty term for misclassification.

There are also non-linear variants of SVM which can be used to classify non-linearly separable data. These variants are based on what is commonly known as the “kernel trick", mapping the training samples using a non-linear mapping operator to another space where they could potential be linearly separable and then applying the same linear algorithm in that space. In people detection most of the focus has been on linear SVMs as they enjoy both faster training and classification speeds; consequently, only linear SVMs are considered as weak learners with the proposed cascaded detector.

3.5 Pareto-Front Analysis

Recall that the total number of weak learners or features considered is 3360. As it will become evident in section 3.6 these amount of features are too much for a tractable discrete optimization and hence the number of feature must be pre-reduced. To do this, we propose to use Pareto Front Analysis by retaining only dominant features—based on their TPR (which is 1-MR), FPR, and computation time.

Pareto-Front analysis deals with selecting the optimal solutions when faced with competing multi-objective optimization criteria, like in equation 3.2 where the objective is to minimize MR, FPR, and Computation Time (CTx). It is termed competing because one has to worsen the other objectives to improve itself. The optimal solutions for these kind of optimization are termed as the Pareto optimal solutions. Pareto-Front analysis is used to find these optimal solutions that make up the Pareto optimal set—the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest. By exactly using this concept, the subset of features that are Pareto optimal with respect to MR, FPR, and CT, are extracted to be used for the discrete optimization step. In this framework, a feature $\mathbf{x}$ is said to *dominate* another feature $\mathbf{x}'$, only if $g_l(\mathbf{x}) \leq g_l(\mathbf{x}')$ for all $l \in \{MR, FPR, CT\}$, and $g_l(\mathbf{x}) < g_l(\mathbf{x})$ for at least one l. Here, the $g_l(\cdot)$ denotes the MR, FPR, or CT, determined using the weak learner trained with the corresponding feature $\mathbf{x}$ and evaluated on a validation set. On the other hand, a feature is said to be *non-dominated* if no other feature *dominates* it. The Pareto optimal features set, $\tilde{\mathcal{F}}$, is actually the set that fulfills equation 3.3. Since, we are dealing in discrete space, this set, $\tilde{\mathcal{F}}$, can easily be determined using algorithm 3.1. Figure 3.6 illustrates a computed Pareto-Front, in 3D space and projected (MR vs CT and FPR vs CT) 2D plot.

$$\begin{aligned} &minimize \quad g(\mathbf{x}) = [g_{MR}(\mathbf{x}), g_{FPR}(\mathbf{x}), g_{CT}(\mathbf{x})] \\ &s.t. \qquad\qquad\quad \mathbf{x} \in \mathcal{F} \end{aligned} \tag{3.2}$$

$$\tilde{\mathcal{F}} = \{\mathbf{x} \in \mathcal{F} \mid \forall_{\mathbf{x}' \in \mathcal{F}} \quad g_{MR}(\mathbf{x}') \geq g_{MR}(\mathbf{x}) \vee g_{FPR}(\mathbf{x}') \geq g_{FPR}(\mathbf{x}) \vee g_{CT}(\mathbf{x}') \geq g_{CT}(\mathbf{x}) \quad \} \tag{3.3}$$

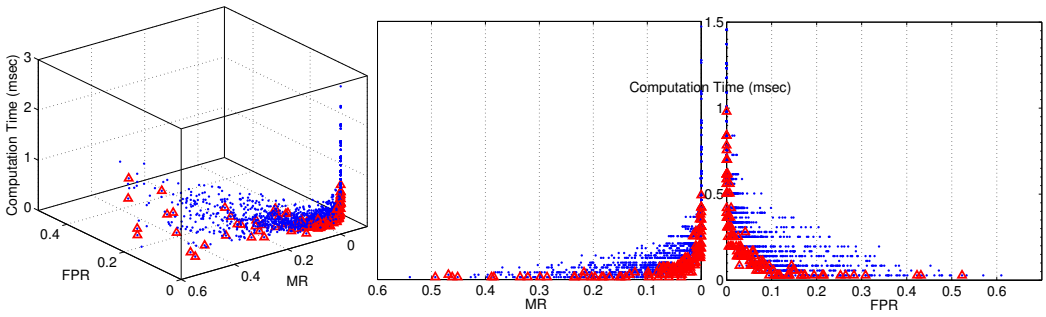


Figure 3.6: Sample extracted Pareto Front. Each of the 3360 features are plotted as a blue dot using their MR, FPR, and CT values. The extracted dominant features (that make the Pareto Front) are shown with red triangular markers. The plot is shown in 3D as well as projected 2D plots to aid visualization.

Algorithm 3.1 Pareto-Front Computation

```
1:  procedure PARETO_FRONT(F)
2:      F̃ ← F_1
3:      for each x ∈ F do
4:          add ← true
5:          for each x' ∈ F̃ do
6:              if x dominates x' then
7:                  F̃ ← F̃\{x'}; continue;
8:              else if x' dominates x then
9:                  add ← false; break;
10:             end if
11:         end for
12:         if add then F̃ ← F̃ ⋃ {x}
13:     end for
14:     return F̃
15: end procedure
```

3.6 Feature Selection via Binary Integer Programming

In this section, the formulation of the discrete optimization based on Binary Integer Programming used for features selection is presented. Initially there were a total of $\mathcal{F}$ features

(and associated weak learner). These number of features have been substantially decreased to smaller set $\tilde{\mathcal{F}}$ using Pareto-Front analysis. In this step, only a few of these features are retained, $\hat{\mathcal{F}} \subseteq \tilde{\mathcal{F}}$. As discussed in the framework presentation, BIP is a special case of integer programming where decision variables are required to be 0 or 1 (rather than arbitrary integers). It aims at minimizing a given linear objective function $f = c^{\intercal}\mathbf{v}$ subject to the constraints that $A.\mathbf{v} \geq b$, where $\mathbf{v}$ represents the vector of 0-1 variables (to be determined), c and b are known coefficient vectors, A is a matrix of coefficients (called constraint matrix). It is well-known that BIP is $\mathcal{NP}$-hard in the strong sense but, in practice, branch-and-cut techniques are able to solve huge binary integer program very fast [Roy 1986, Wolsey 2003]. In this work, BIP is used to select a subset of features that fulfill the detection performance stipulated (in terms of TPR and FPR) with the minimum combined computation time. With respect to the reduced feature set, $\hat{\mathcal{F}}$, the BIP solutions are going to be the optimal choices.

The BIP formulation for feature selection is presented as follows.

Definition of parameters: The following are list of parameters used in the optimization specification along with their definitions. For clarification, a binary set is denoted as $\mathbb{B} = \{0, 1\}$.

- $N = \{1, ..., n\}$: set of training sample indexes with $n \in \mathbb{Z}$; a total of n training samples indexed by i;
- $M = \{1, ..., m\}$: set of weak learners indexes with $m \in \mathbb{Z}$; a total of m weak learners indexed by j;
- $\mathbf{y}^+ \in \mathbb{B}^n$, $\mathbf{y}^+ = \left\{y_i^+\right\}_{i \in N}$; $\mathbf{y}^- \in \mathbb{B}^n$, $\mathbf{y}^- = \left\{y_i^-\right\}_{i \in N}$; notice $y_i^- + y_i^+ = 1 \quad \forall i \in N$

$$y_i^+ = \begin{cases} 1 & \text{if } i \text{ is positive} \\ 0 & \text{else} \end{cases} \qquad y_i^- = \begin{cases} 1 & \text{if } i \text{ is negative} \\ 0 & \text{else} \end{cases}$$

- $\mathbf{H} \in \mathbb{B}^{nxm}$ where $\mathbf{H} = \{\mathfrak{h}_{i,j}\}_{\substack{i \in N \\ j \in M}}$ with $\mathfrak{h}_{i,j} \in \{0, 1\}$

$$\mathfrak{h}_{i,j} = \begin{cases} 1 & \text{if weak learner } \mathfrak{h}_j \text{ detects sample } i \text{ as positive} \\ 0 & \text{else} \end{cases}$$

- TPR $\in [0, 1]$: minimum true positive rate set at the considered node of the cascade;
- FPR $\in [0, 1]$: maximum false positive rate at the node;
- $\tau \in \mathbb{R}^m$: with $\tau = \{\tau_j\}_{j \in M}$ computation time of weak learner j.

Decision Variables: In BIP, the decision variables are restricted to binary values, values from the set $\mathbb{B} = \{0, 1\}$. The BIP decision variables are the following.

- $\mathbf{v} \in \mathbb{B}^m$, $\mathbf{v} = \{v_j\}_{j \in M}$ $v_j \in \{0, 1\}$: $v_j = 1$ if weak learner $\mathfrak{h}_j$ is selected, else $v_j = 0$;
- $\mathbf{t} \in \mathbb{B}^n$, $t_i \in \{0, 1\}$: $t_i = 1$ if a positive sample i has been detected as positive (true positive) by at least one selected weak learner, else $t_i = 0$;
- $\mathbf{f} \in \mathbb{B}^n$, $f_i \in \{0, 1\}$: $f_i = 1$ if a negative sample i has been detected as positive (false positive) by at least one selected classifier, else $f_i = 0$.

Let vector $\mathbf{p} = \{p_i\}_{i \in N} = \mathbf{Hv}$, which denotes the total number of weak learners that have labeled each training sample i as positive.

Objective Function and Constraints:

$$
\begin{array}{lll}
\min & \tau^{\top}\mathbf{v} & (1)\\
\text{s.t} & t_i \leq y_i^{+} \cdot p_i \qquad \forall i & (2)\\
& f_i \geq y_i^{-} \cdot \mathfrak{h}_{i,j} \cdot v_j \qquad \forall (i,j) & (3)\\
& \|\mathbf{T}\|_1 \geq \|\mathbf{y}^{+}\|_1 \cdot \mathrm{TPR} & (4)\\
& \|\mathbf{F}\|_1 \leq \|\mathbf{y}^{-}\|_1 \cdot \mathrm{FPR} & (5)\\
& \mathbf{v} \in \mathbb{B}^n; \mathbf{T} = \{t_i\}_{i\in N}, \mathbf{F} = \{f_i\}_{i\in N}; \mathbf{T}, \mathbf{F} \in \mathbb{B}^n & (6)\\
& \|\cdot\|_1 \text{ is } l_1 \text{ norm.}
\end{array}
$$

The objective function (1) aims at minimizing the computation time. Constraints (2)-(5) express that a given rate of detection quality has to be reached (depending on the number of true and false positives). Constraints (2) link v_j and t_i variables (via p_i) so that $t_i = 0$ if image i has not been well-recognized by at least one selected classifier. Constraints (3) link v_j and f_i variables so that $f_i = 1$ if a negative image i has been recognized as positive by at least one selected classifier. Constraint (4) expresses that the rate TPR of true positives, obtained with the selected classifiers, has to be reached. Similarly, constraint (5) expresses that the rate FPR of false positives, obtained with the selected classifiers, must not be exceeded. In this formulation, there are a total of $(n \cdot (m+1)+2)$ number of constraints, which could be huge for large n and m values. The final subset of features $\hat{\mathcal{F}}$ corresponds to only the selected features, *i.e.*, non zero $\mathbf{v}$ entry; since each feature indexed by j is associated with a unique weak learner $\mathfrak{h}_j$, $\hat{\mathcal{F}}$ also represents the subset of weak learners retained.

3.7 Discrete AdaBoost and Cascade Construction

In the previous section, the discrete optimization formulation that selects pertinent features with the minimum possible computation time has been presented. The next step is to use these subset of features to build a composite per nodal strong classifier. In the literature, the ideal candidates for this task are Boosting variant classifiers. Specifically, AdaBoost has been highly credited for this [Schapire 2003]. AdaBoost is an algorithm for constructing a "strong" classifier as linear combination of weaker classifiers. Of all the Boosting variants, we have chosen to use Discrete AdaBoost because of the following reasons: (1) the discrete nature of our feature selection procedure, (2) Discrete AdaBoost's simplicity, and (3) its proven performance.

Here below, the Discrete AdaBoost algorithm along with the final complete cascaded classifier construction procedure are presented.

Discrete AdaBoost Discrete AdaBoost is one instance of the Boosting classifier variants which build a strong classifier as linear combination (weighted voting) of a set of weak classifiers. Suppose, we have a labeled training set $\{(x_i, y_i)\}_{\{i=1,...,(n_+ + n_-)\}}$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$, where n_+ and n_- denote the number of positive and negative training samples respectively. Given a set of weak learners (features) $\hat{\mathcal{F}} = \{\mathfrak{h}_j\}_{j\in M}$, with $M = \{1, 2, ..., m\}$ the total number of weak learners, that can assign a given example a corresponding label, *i.e.*, $\mathfrak{h} : x \rightarrow y$, Discrete Adaboost constructs a strong classifier of the form $\mathcal{H}(x) = \sum_{t=1}^{T} \alpha_t \mathfrak{h}_t(x)$ with $sign(\mathcal{H}(x))$ determining the class label. The t indexes connote the sequence of the weak learners and this specific classifier has a total of T weak learners. The specific weak learner to use at each iteration of this boosting algorithm and the associated weighting coefficients, α_t, are derived minimizing the exponential loss, which provides an upper bound on the actual 1/0

loss [Schapire 2003]. The complete algorithm for training a per node classifier with Discrete AdaBoost is detailed in algorithm 3.2.

Algorithm 3.2 Discrete AdaBoost Training

procedure TRAIN_ADABOOST($\hat{\mathcal{F}}$, $\{(x_i, y_i)\}_{i \in N}$)
  Initialize: $D_1(i) = \dfrac{1}{(n_+ + n_-)}$ $\quad$ *\\distribution over training samples; $(n_+ + n_- = n)$*
  for t = 1,2,...,T **do**
    Find the best weak learner $\mathfrak{h}_t$:
      $\rightarrow \mathfrak{h}_t = \arg\min_{\mathfrak{h}_j \in \hat{\mathcal{F}}} \epsilon_j$ where $\epsilon_j = \sum_{i=1}^{n_+ + n_-} D_t(i)[y_i \neq \mathfrak{h}_j(x_i)]$
    Compute weak learner weight: $\alpha_t = \dfrac{1}{2} \ln \dfrac{1 - \epsilon_t}{\epsilon_t}$
    Update data weight distribution: $D_{t+1}(i) = \dfrac{D_t(i) \exp(-\alpha_t y_i \mathfrak{h}_t(x_i))}{Z_t}$
    \\Z_t is a normalization factor used to make D_{t+1} a distribution
  end for
  Strong classifier: $\mathcal{H}(x) = \sum_{t=1}^{T} \alpha_t \mathfrak{h}_t(x)$
  Computation time of cascade node: $\zeta = \sum_{t=1}^{T} \tau_t$
  return $\mathcal{H}(\cdot)$ and ζ
end procedure

Cascade Construction The presented Discrete AdaBoost algorithm is used to build a strong classifier per node. The final classifier used for detection however contains multiple nodes forming a cascade. The complete cascade structure of the final detector is built at the end of the training process. The cascade construction process employed is trivial and relies upon the works of Viola and Jones [Viola 2004]. It starts with all positive training samples and a subset of the negative training samples (equivalent to the positive ones) to learn the set of relevant features and classifier for the initial cascade node. Once this is done, all negative training samples in the dataset are tested with it. All those that get classified correctly are rejected while all those labeled as positive samples (false positives) are retained along with the positive samples for training the following nodes. This step is repeated until the negative training sample repository gets depleted. This data mining technique makes it possible to use vast number of negative training samples.

3.8 Experiments and Results

In this section all experiments carried out to evaluate the performance of the proposed framework are presented in detail. The experiments are focused on two main aspects—detection performance, and computation time—of the proposed detector and comparison with the prevalent approaches in the literature.

The important investigations carried out can be summarized accordingly:

- Ladybug dataset - the experiments on the Ladybug dataset include, training and testing a detector based on: (1) the presented framework using Fisher's LDA and a decision tree as a weak learner, (2) the presented framework using linear SVM as a weak learner, and (3) Dalal and Triggs implementation of HOG based detector [Dalal 2005]. Detection performance are quantified using Per Window based DET metrics presented in section 2.4.1.
- INRIA dataset - experiments similar to the ones carried out using the Ladybug dataset are carried out. In addition, Full Image evaluations (section 2.4.3) are carried out to compare the performance with the state-of-the-art people detection approaches presented in chapter 2.
- Caltech dataset - on the Caltech dataset, only testing experiments using the best detector model trained on the INRIA dataset, with the proposed framework, are carried out. Comparative performance evaluation is again carried out using the Full Image evaluation metrics.

The detection framework proposed in this chapter tries to take both detection performance and computation time into consideration. As a result, it is imperative to determine and report its overall computation time and how its speed compares with the existing methods in the literature. For that, we have used the average speed up measure in equation 3.4. The average speed up reports the average computation time taken by the proposed cascaded detector relative to Dalal and Triggs detector [Dalal 2005]. Clearly, the number of person containing candidate windows are relatively very small compared to the number of total candidate windows generated from person free zones. Hence, the total number of windows tested by cascade is highly influenced by the FPR. This means, assuming a constant FPR for all the nodes of the cascade, if there are N_w candidate windows, it is safe to assume only N_w·FPR windows will pass onto the next stage and on average the k^{th} node will only evaluate $N_w \cdot (\text{FPR})^{(k-1)}$ candidate windows. With this, if the total computation time taken by node k to evaluate a single candidate window is represented by ζ_k, the average computation time for a cascade with K nodes, ζ_K, is: $\zeta_K = \sum_{k=1}^{k=K} N_w \cdot (\text{FPR})^{(k-1)}.\zeta_k$. If we represent the time taken by Dalal and Triggs HOG to evaluate a single candidate window to be ζ_{HOG}, the total computation time to evaluate N_w windows is $N_w \cdot \zeta_{HOG}$. This leads to an average speed up term, with respect to Dalal and Triggs, given by equation 3.4.

$$\text{Average Speed Up} = \frac{\zeta_{HOG}}{\sum_{k=1}^{k=K} (\text{FPR})^{(k-1)} \cdot \zeta_k} \tag{3.4}$$

But, recall that ζ_{HOG} and ζ_k are both integral multiples of the computation time taken to extract and evaluate a single HOG feature block. This simplifies the computation further and it becomes a ratio of number of constituent HOG feature blocks weighted by the cumulative FPR in the denominator.

With this experimental setup in mind, the following subsections discuss implementation details along with different validation runs performed, and results obtained on the proprietary and public datasets presented in chapter 2.

3.8.1 Implementation Details and Validation

From the software perspective, the complete framework is implemented using the C++ programming language. This is done for speeding up training and testing periods. In addition this will ease further integration on robotic platforms for real time demonstration and experiments. When ever possible the framework relies on reputed scientific libraries to ease better

reproducibility, understandability, and credibility by other researchers. The specific open source libraries the presented framework relies on are listed in table 3.2. The complete framework, including the remaining components which are custom developed, is developed following good software development practices.

Table 3.2: Open source libraries used for implementing different components of the proposed framework.

Functions/Algorithms/Tasks	Software Library
Fisher's LDA	ALGLIB library [ALGLIB]
Linear SVM	SVMlight library [Joachims 1999]
Discrete optimization solver	Gurobi Optimizer [Gurobi 2013]
Decision tree	OpenCv [OpenCv]
Image input/output	OpenCv [OpenCv]

3.8.1.1 Cascade Node

The cascade node construction is governed by two provided parameters: the nodal TPR and FPR. The training is carried out in such a way that the final trained classifier conforms to these stipulated performance requirements. Each cascade node is built using a subset of the total negative training samples and all positive samples. In all cases, equal number of negative samples as that of the positive samples are used for each cascade node, *i.e.*, 1990 positive and negative windows for the Ladybug dataset and 2416 positive and negative windows for the INRIA dataset. This set is initially divided into a 60% training and a 40% validation set. The weak learners are trained using the 60% training set. Then, TPR and FPR values corresponding to each weak learner are determined based on the validation set. All subsequent computation, *i.e.*, Pareto-Front analysis and feature selection via discrete optimization are performed using the weak learners performance conferred on the validation set. Once the pertinent features are selected, the corresponding weak learners are re-trained using the combined training and validation set within the Discrete AdaBoost to build the per node final strong classifier, *i.e.*, $\mathcal{H}(\cdot)$.

Weak learners In this framework, two different types of weak learners are considered. The first one is linear SVM. At each node of the cascade, this classifier is trained initially using 60% of the training set and then the complete 100% once the feature selection and validation steps are finalized. This classifier needs no further processing as it furnishes a class label and confidence score determined by the distance from the classifying hyperplane.

On the other hand, the second type of weak learner considered, Fisher's LDA, only provides a projecting vector that maximizes the separability between the two classes; further classification of the corresponding scalar projected values is necessary. As discussed previously, we have chosen to use decision trees built on the scalar values to do the final classification. But, here, it is possible to use a decision tree of any arbitrary depth (multiple partitions of the 1-D space). The higher the depth of the tree, the more accurate it gets to correctly label the given data, but this does not mean it will perform better on test (unseen) data. To validate a proper depth to use for the decision trees, we built a complete cascaded detector using 60% of the training data from the Ladybug dataset with a nodal requirement of 1.0 for TPR and 0.5 for FPR (these values are used on all nodes) and evaluated the performance on the remaining unseen 40% data with varying decision tree depths. Corresponding error plot, cascade node vs error rate ($\frac{\text{no. of correct}}{\text{total tested samples}}$), is shown in figure 3.7. Clearly, a decision tree of depth 2 has better generalization and hence is

chosen for future use. For each feature, the Fisher's LDA projection vector is trained once for the first cascade node and used throughout by only retraining the associated decision trees. It is observed that computing Fisher's LDA vector per each node makes the classifier over-fit on the training set leading to a very deteriorated performance on the validation set. As a result, in the complete training, Fisher's LDA vectors are computed at the initial cascade node and only the associated decision trees of depth 2 are retrained on consecutive cascade nodes.

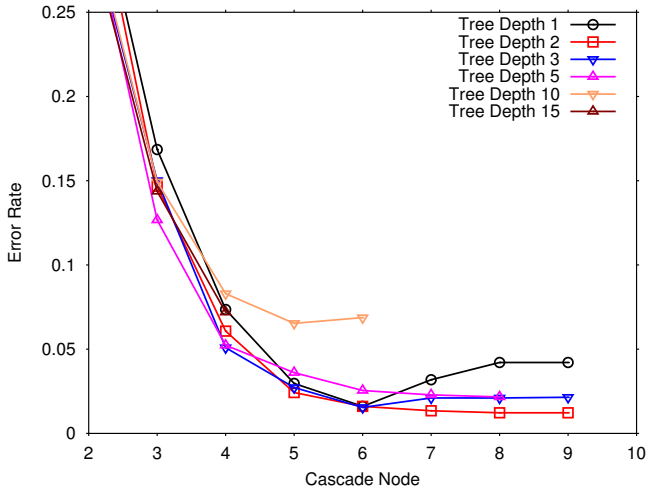


Figure 3.7: Decision Tree depth validation when used in conjunction with Fisher's LDA.

3.8.1.2 Full Cascade Construction

The full cascade construction is carried out as previously explained in section 3.7. If there are K nodes in the full cascade and each k^{th} node has detection performance indicated by TPR_k and FPR_k, the full TPR of the cascade is given by $\text{TPR}_{full} = \prod_{k=1}^{K} \text{TPR}_k$ and the corresponding false positive rate is give by $\text{FPR}_{full} = \prod_{k=1}^{K} \text{FPR}_k$. Evidently, setting a TPR rate of 0.95 for each node will result in a 0.77 TPR_{full} just after five cascade nodes. On the hand setting a higher FPR rate of, for example, say 0.6 for each node will result in an overall FPR_{full} of 0.078 after just five nodes which is actually good. Hence, in light of this, a strict per node TPR of 1.0 is used for all cascade nodes during training, but for FPR values between 0.4 and 0.6 are investigated on the Ladybug dataset and the value that leads to a better detector (as a compromise between detection and speed) is further used during training on the INRIA dataset. In case the strong nodal detector does not achieve the stipulated TPR on the training set by default, the AdaBoost threshold is lowered until a 100% TPR is achieved. This of course might increase the nodal FPR, but it is better to increase the false positives at that specific node, as this will be corrected in further stages of the cascade, rather than lowering TPR which is impossible to correct at later stages.

3.8.1.3 Non-maximal Suppression

The full image evaluation, as will be presented on the INRIA and Caltech datasets in the next subsection, relies on evaluating the final detector performance on a given full realistic image. This entails applying the trained detector in a sliding window mode over all possible positions

and scales of the tested image. This will eventually lead to repeated evaluations of the same areas shifted and/or scaled slightly giving rise to either multiple correct detections or multiple false alarms which could distort reported performance. To alleviate this, it is necessary to apply a non-maximal suppression so that detections arising from the same area will be merged into a single one.

In this implementation, we have privileged a proven technique based on Pairwise Max (PM) suppression [Dollár 2012]. The PM approach suppresses the less confident of every pair of detections that overlap sufficiently. The adopted overlap measure between two detection bounding boxes, R_1 and R_2, is given by $o = \frac{\text{area}(R_1 \cap R_2)}{\min(\text{area}(R_1), \text{area}(R_2))}$. If this o is above a given threshold, the less confident of the two rectangles is discarded and the process continues until no two bounding rectangles with above threshold overlap are left. Figure 3.8 shows detector performance (in terms of log-average miss rate presented in section 2.4.3) for a detector trained only on 50% of the INRIA training dataset and tested on the first 100 images of the full image INRIA test images. The best performance (lowest average miss rate) is obtained for $o = 0.65$, and hence this value is the actual value used for all subsequent non-maximal suppression steps during full image evaluation.

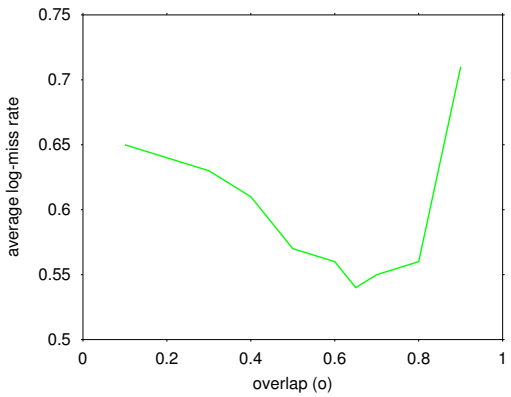


Figure 3.8: Miss rate variation as a function of the detection window overlap for non-maximal suppression.

3.8.2 Results

The experimental results reported are categorized in three headings according to the envisaged experimental investigations. Each category relates to the different datasets used: the Ladybug dataset, the INRIA public dataset, and the Caltech dataset.

3.8.2.1 Ladybug Dataset

The first set of experiments are performed on the Ladybug proprietary dataset. Here, three different classifiers are trained using the Ladybug training set and tested using the corresponding Ladybug test set. The three classifiers are: (1) the proposed framework using Fisher's LDA and a decision tree of depth 2 as a weak learner (shortened as BIP with LDA+DT), (2) the proposed framework using linear SVM as a weak learner (referred as BIP with SVM), and (3) Dalal and Triggs HOG detector explicitly trained on the Ladybug training dataset. For the BIP based variants, three different detectors using a nodal FPR of 0.4, 0.5, and 0.6 with a 1.0 TPR are trained. The obtained results are summarized in table 3.3 and corresponding DET curves are shown in figure 3.9.

Generally, higher values of FPR result in, relatively speaking, simpler features in the initial stages leading to a reduced computation time (higher average speed up), but this is achieved at the expense of reduced detection performance. At 10^{-4} FPPW, the best detection performance is obtained using the proposed framework trained with SVM weak learners and a nodal FPR of 0.4, a marked 1.7% detection improvement and an average speed up of 6.92x over Dalal and

Table 3.3: Comparative summary of learned cascade classifiers on Ladybug dataset with varying FPR and Dalal and Triggs detector.

Detector (with associated weak learner)	Per Node FPR	No. of features	K (no. of nodes)	Average speed up over [Dalal 2005]	Miss rate at 10^{-4} FPPW
BIP with SVM	0.4	9	5	6.92x	**0.010**
BIP with SVM	0.5	14	7	8.09x	0.025
BIP with SVM	0.6	27	10	8.86x	0.134
BIP with LDA+DT	0.4	23	6	8.72x	0.011
BIP with LDA+DT	0.5	42	9	9.22x	0.029
BIP with LDA+DT	0.6	**43**	**11**	**9.68x**	0.128
[Dalal 2005]	–	1	1	1.0x	0.027

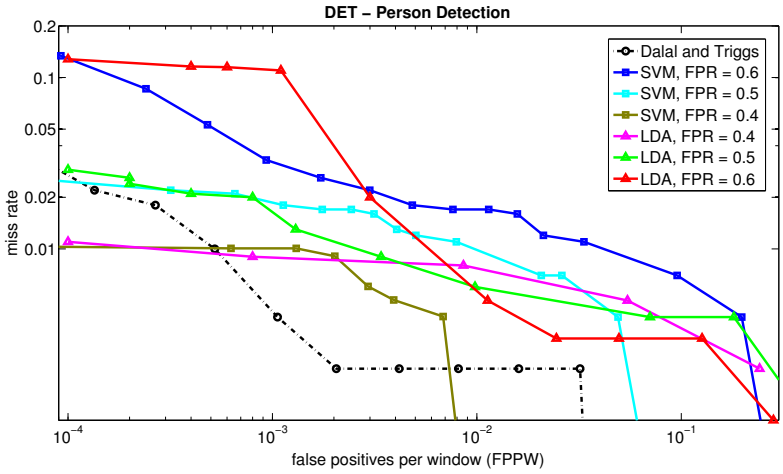
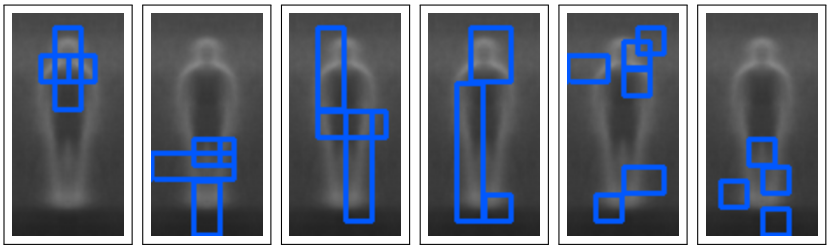


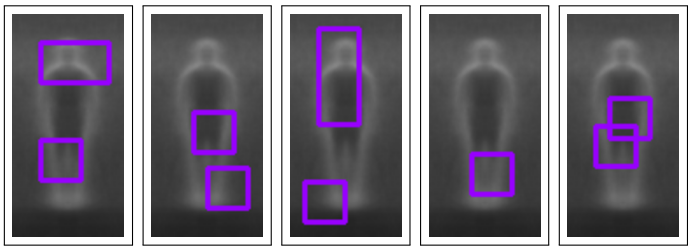
Figure 3.9: DET plots showing performance on Ladybug test dataset. SVM and LDA refer to detector trained using the proposed framework with SVM based and Fisher's LDA+DT based weak learners consecutively.

Triggs detector. In fact, it is also possible to get a 9.68 average speed up with only a 10% loss in Miss Rate at 10^{-4} FPPW which could be acceptable for systems with a tight running speed constraint. At higher FPPWs, Dalal and Triggs detector does considerably well, but as one progresses along the cascade of the proposed framework, more and more false positives are rejected eventually catching up in performance with maintained consistent speed improvement.

Figure 3.10 shows the selected features using the proposed framework and a nodal FPR of 0.4. The features are shown overlaid on an average gradient image determined by averaging the gradient images of all positive training windows. In the initial node, the features selected using SVM have higher spatial support, compared to the LDA variants, which explains the increased computation time (hence, reduced speed up). In both cases, the selected features conform to the salient gradient regions and seem to capture discriminatory cues common to people in general.



(a) Selected features at each node of the cascade trained using Fisher's LDA+DT weak learner and a nodal FPR of 0.4



(b) Selected features at each node of the cascade trained using SVM weak learner and a nodal FPR of 0.4

Figure 3.10: Illustration of selected features overlaid on an average gradient image.

3.8.2.2 INRIA Dataset

Tests on this dataset are carried out to see the performance of our cascaded detector on a public dataset and eventually compare its performance with Dalal and Triggs (and other state-of-the-art detectors) given the dataset has a lot of intra-class and inter-class variation. Taking the results obtained on the Ladybug dataset into consideration, the cascade detector on the INRIA dataset is trained using an FPR of 0.5 as a compromise between detection performance and obtained speed improvements. Specifically, two cascade detectors based on the proposed framework—with Fisher's LDA+DT and linear SVM as weak learners—are trained with the INRIA training dataset. The obtained results are summarized in table 3.4 and figure 3.11 depicts the DET plot.

Table 3.4: Comparative summary of learned cascade classifiers on INRIA dataset, using a constant per node FPR of 0.5 and TPR of 1.0, and Dalal and Triggs detector.

Detector (with associated weak learner)	No. of feature vectors	K (no. of nodes)	Average speed up over [Dalal 2005]	Miss rate at 10^{-4} FPPW
BIP with SVM	58	**13**	2.21x	0.123
BIP with LDA+DT	50	8	**2.46x**	0.275
[Dalal 2005]	1	1	1.0x	**0.109**

The detector built using the proposed framework and SVM as weak learner has 13 cascade nodes achieving a 0.123 Miss rate at 10^{-4} FPPW. It has an average speed up of 2.21x compared to Dalal and Triggs detector. This speed up is achieved with only a 1.4% reduction in the miss rate at the specified FPPW. Between the 10^{-2} to 10^{-4} FPPW interval, the miss rate is quite comparable with only marginal loss.

In this experiment, there is a marked difference in detection performance between LDA

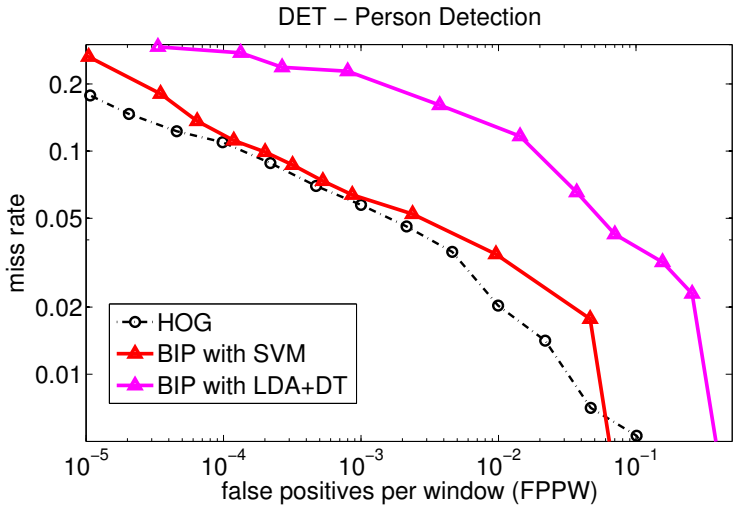
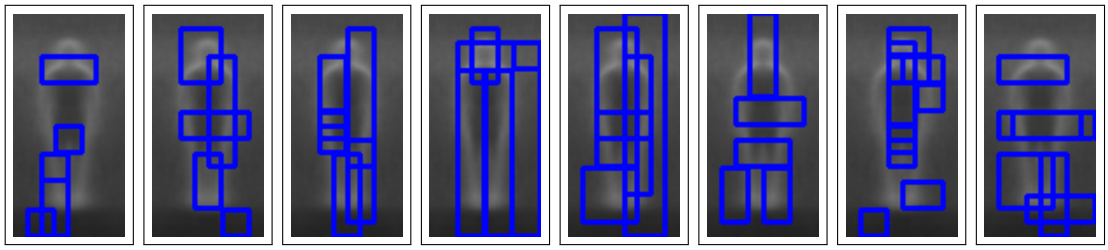
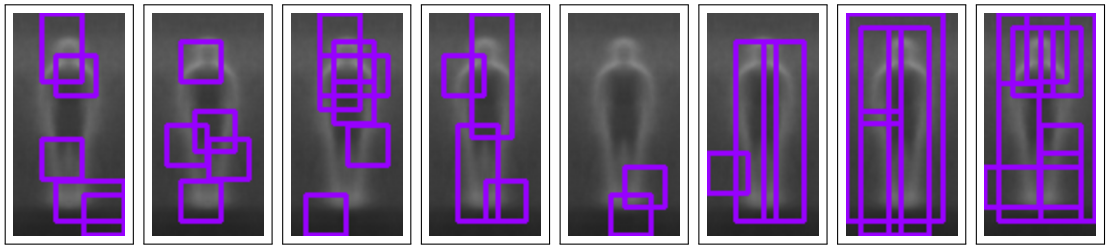


Figure 3.11: Comparative curve for selected cascade detector and Dalal and Triggs detector on the Ladybug dataset.

based and SVM based detector at 10^{-4} FPPW. This arises due to the increased variability in the training set that makes it difficult for the LDA+DT weak learner to do as well as it did on the Ladybug dataset. Nonetheless, the LDA+DT based detector performs good achieving a miss rate of 0.28 at 10^{-5} FPPW, which is comparable to the SVM variant at this FPPW, and an average speed up of 2.46 over Dalal and Triggs. The features selected in the first 8 nodes of the two cascaded detectors are shown in figure 3.12.



(a) Selected features using Fisher's LDA+DT weak learners.



(b) Per node selected features using SVM weak learners.

Figure 3.12: Selected features of the detectors trained on the INRIA dataset. In both cases a constant nodal FPR of 0.5 is used.

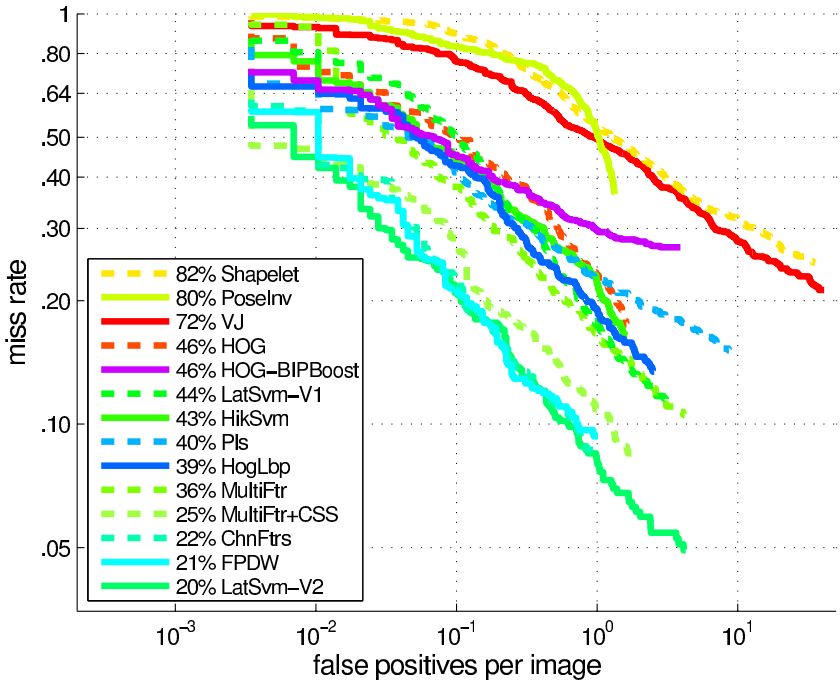


Figure 3.13: Full image evaluation results on the INRIA test dataset. For all the other approaches the results published in [Dollár 2012] are used.

To compare the performance of our best trained detector with respect to the state-of-the-art approaches in the literature, we have evaluated it on the INRIA dataset using the full image evaluation metrics. This enables direct comparison with already published results. We evaluate the cascaded detector variant trained with linear SVM as a weak learner with nodal FPR of 0.5. It is referred as **HOG-BIPBoost** and corresponding results are shown in figure 3.13. For non-maximum suppression an overlap threshold of 0.65 is used. The detector performances are summarized using the log-average miss rate metrics, which represents the area under the curve. Our HOG-BIPBoost detector achieves a log-average miss rate of 46% which is equivalent to that achieved by Dalal and Triggs HOG detector. It also shows superior performance compared to **VJ**, **Shapelet**, and **PoseInv** detectors. Computation wise, **HOG-BIPBoost** achieves the 3^{rd} best speed next to **ChnFtrs** and **LastSvm-V2**. Bear in mind, this speed is determined by the model trained on the INRIA dataset. The models trained on the Ladybug dataset actually achieve a minimum of **1.66 fps** (BIP with SVM trained with nodal FPR of 0.4) and a maximum of **2.32 fps** (BIP with LDA+DT trained with nodal FPR of 0.6).

3.8.2.3 Caltech Dataset

To further evaluate the trained models on a very challenging dataset, further evaluations are performed on the Caltech dataset. In this step, we have used the best cascade detector, *i.e.*, BIP with SVM weak learners, trained on the INRIA dataset to test on the Caltech test set. Due to the huge amounts of data in this dataset, the tests are carried out using every 30^{th} image frame with the help of the Matlab toolbox from [Dollár 2012]. In total 4286 image frames of 640×480 are used. The evaluations, shown in figure 3.14, are categorized accordingly: *Near scale*, people which are at least 80 pixels tall; *Medium scale*, people between 30 to 80 pixels long;

Table 3.5: Computation time comparison with the state-of-the-art. The values for the different detectors are taken from [Dollár 2012]. These values are determined on a 640 × 480 sized images detecting people with a minimum height of 100 pixels.

Detector	Shapelet	PoseInv	VJ	HOG	HOG-BIPBoost	LatSvm-V1	HikSvm	Pls	HogLbp	MultiFtr+CSS	ChnFtrs	LatSvm-V2
Fps[a]	0.05	0.47	0.45	0.24	0.53	0.4	0.19	0.02	0.06	0.03	**1.18**	0.63

[a]Run times of all detectors are normalized to the rate of a single modern machine [Dollár 2012].

No occlusion, all people without any occlusion (fully visible); *Partial occlusion*, with less than 35% of their body occluded; *Reasonable*, people over 50 pixels tall and with no occlusion what so ever; and *Overall*, taking all annotated people into consideration.

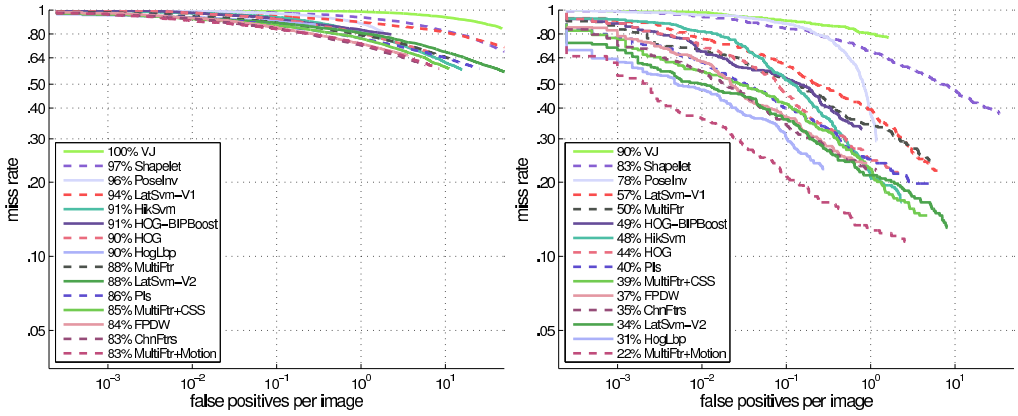
In all categories, the HOG-BIPBoost detector trails Dalal and Triggs HOG hand in hand with a maximum of 6% log-average miss rate reduction in partially occluded people. Overall performance, it trails by 8% compared to the best approach and only 1% compared to Dalal and Triggs HOG with a 91% log-average miss rate. Like all the other approaches, its best performance is manifested on the near scale people with a 49% log-average miss rate. Clearly, it suffers the most with people of medium scale and under the presence of partial occlusion which down weigh its overall performance.

3.8.3 GPU Implementation

Of all the detector variants we have proposed and evaluated, the detector trained with the Ladybug dataset using LDA+DT as a weak learner at a nodal FPR of 0.6 achieves the best frame rate of 2.32 fps on a 640 × 480 image (normalized to the rate of a single modern machine). And, its counter part that uses SVM and trained on the Ladybug dataset achieves 1.66 fps. Both of these are quite far to be used in any acceptable real time system. To bring this to a usable form, we have implemented a GPU version of the variant that uses SVM as a weak learner utilizing OpenCv's GPU programming interface [OpenCv]. The SVM variant is chosen because it directly integrates with OpenCv's GPU version of Dalal and Triggs [Dalal 2005] detector. On our GPU hardware, an nVidia GeForce GF108 (Quadro 1000M), our detector, which uses SVM weak learners and trained with nodal FPR of 0.4 on the Ladybug dataset, achieves 12.5 fps with 640 × 480 image size. This is also a significant improvement on the GPU version of Dalal and Triggs, which actually achieves 7.7 fps with the same images. In all cases, only people above 100 pixels high are considered.

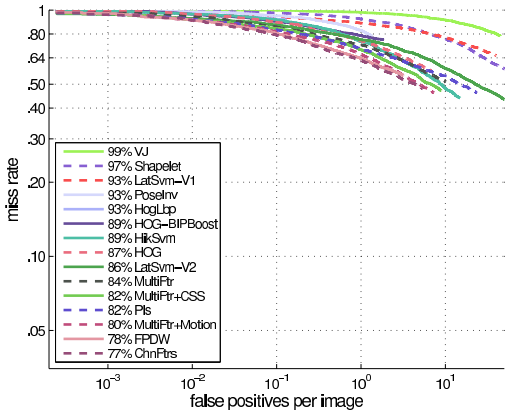
3.9 Discussions

In this chapter we presented a people detection framework based on Viola and Jones [Viola 2004] cascade configuration that showcases a novel feature selection scheme. The framework is oriented on two aspects that are of paramount importance in people detection: detection performance and computation time. The proposed framework tries to find a compromise between the two usually contradicting requirements. Specifically, we focus on HOG feature which has proven to be the

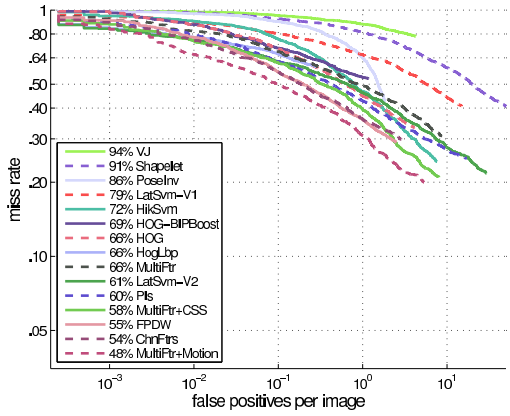


(a) Overall

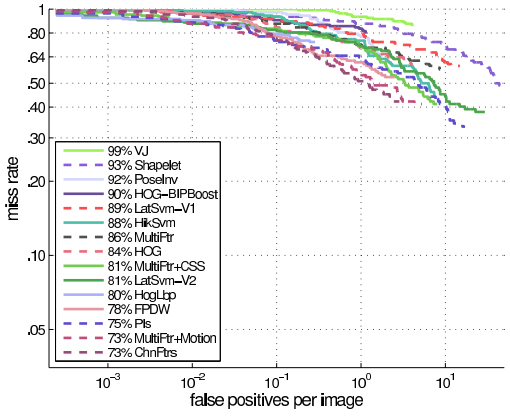
(b) Near scale



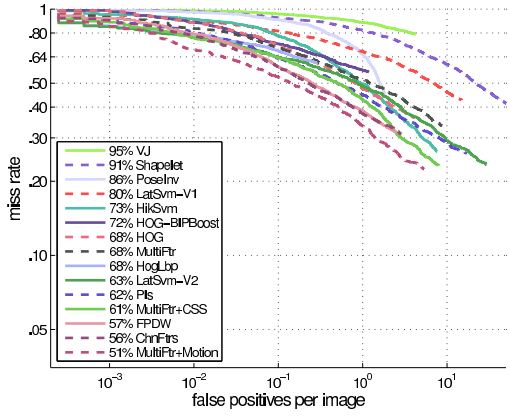
(c) Medium Scale



(d) No occlusion



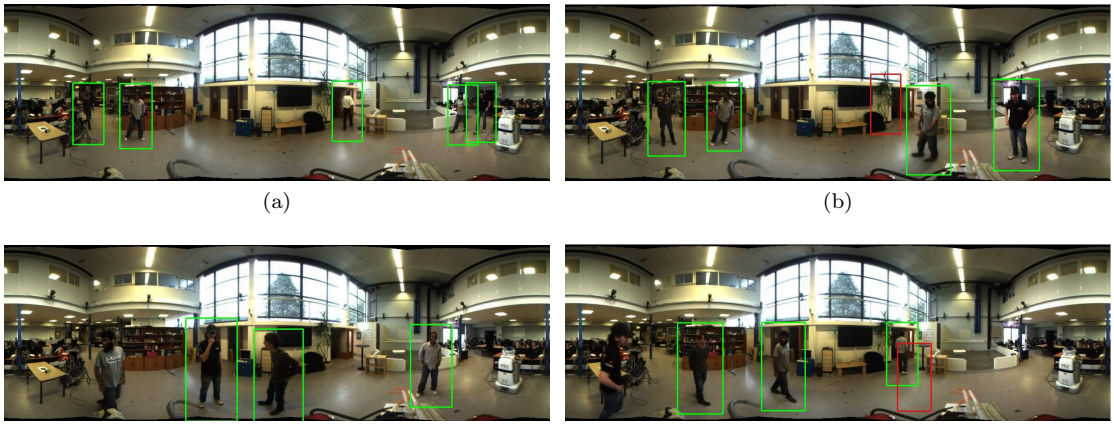
(e) Partial occlusion



(f) Reasonable

Figure 3.14: Full image evaluation results on the Caltech test dataset.

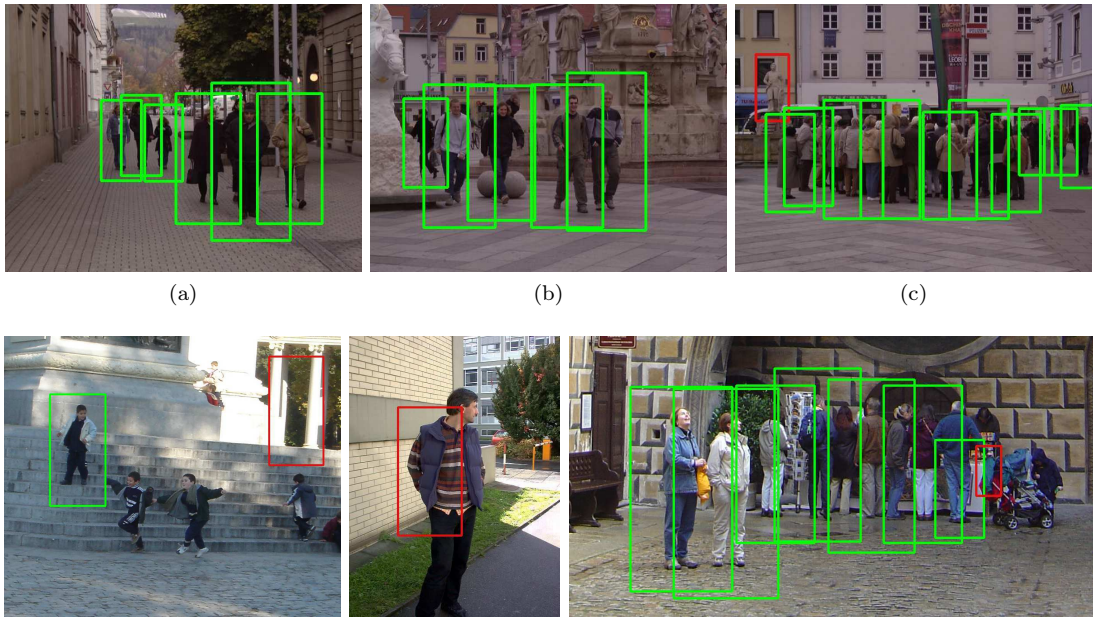
most successful feature for people detection. To tackle the computation time taken with this feature during extraction and classification, we begin by subdividing the high dimensional vector into an over complete set made up of neighboring blocks of the high dimensional vector, which



(a) (b)

(c) (d)

Figure 3.15: Sample full image detection outputs from the Ladybug dataset. Green corresponds to correct detections and red to mistakes.

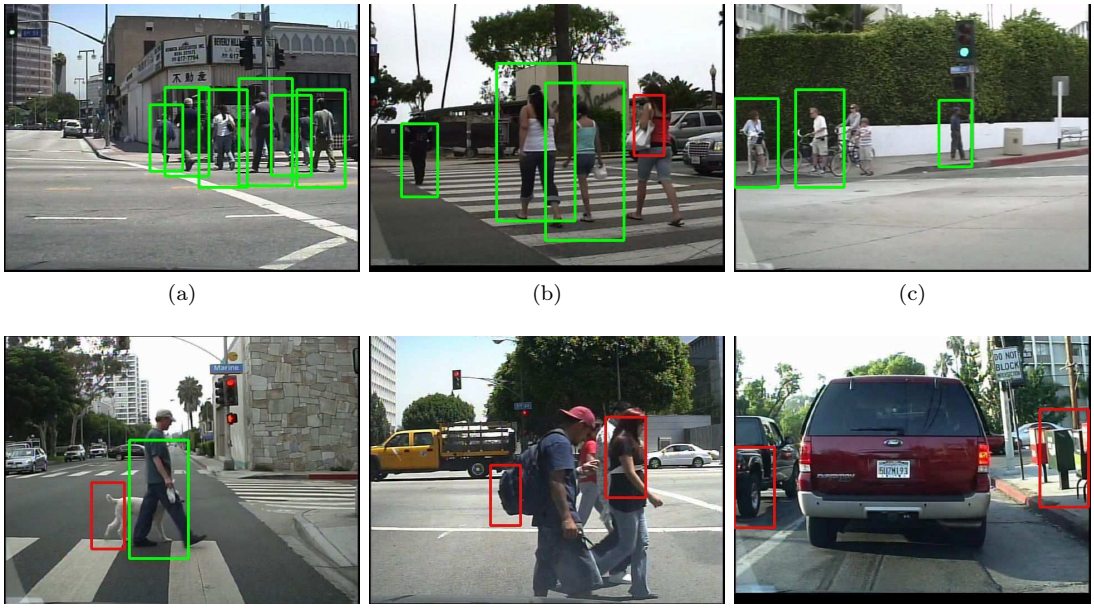


(a) (b) (c)

(d) (e) (f)

Figure 3.16: Full image detection illustrations on images taken from the INRIA test dataset. Green corresponds to correct detections and red to mistakes.

results in a pool of variable dimension features. Then we propose Binary Integer Programming based optimization to select the subset of pertinent features, at each cascade node, that are strictly sufficient to achieve the stipulated detection performance with the minimum combined computation time. In conjunction with AdaBoost, this leads to a systematic construction of a cascaded detector learned from the training data exhaustively taking computation time explicitly into consideration.



(a) (b) (c)

(d) (e) (f)

Figure 3.17: Illustrative full image detections taken from the Caltech dataset. Green corresponds to correct detections and red to mistakes.

The performance of the proposed framework is demonstrated in three different datasets: a proprietary dataset acquired in our robotic laboratory, and two public datasets gathered in different contexts. The initial evaluation focused on the proprietary dataset, the Ladybug dataset, which actually features persons with less pronounced pose variation. In this dataset, out of all 6 detectors trained with varying weak learners and stipulated nodal FPR, it outperformed Dalal and Triggs detector (trained on the same dataset) in 2 cases at 10^{-4} FPPW, and it suffers only a 10% loss in miss rate in the worst case, at the same FPPW. On the other hand, the average speed up ranged from 6.92 to 9.68 times that of Dalal and Triggs detector. This is quite commendable especially as it manages to improve both criteria, detection rate and speed, in 2 out of the 6 trained cases.

Consequently, the following investigations focused on evaluation on a more challenging dataset, the INRIA public dataset. This dataset contains images of people acquired in outdoor and in few occasions indoor public environments. The featured people exhibit more variations in pose situated in differing scenes. The two detectors trained on this dataset resulted in more complex features, compared to the Ladybug dataset counterpart, to capture the high intra-class and inter-class variability. The DET results obtained with the detector trained using SVM as weak learner show superior performance over the one trained using LDA+DT as weak learner. This detector also shows comparable detection performance to that of Dalal and Triggs HOG with a 2.21 times speed improvement over it. In fact, it shows only a 1.4% percent reduction in miss rate at 10^{-4} FPPW with the per window DET metrics and an equivalent log-average miss rate with full image evaluations.

Sample detections on test images from the different datasets are shown in figures 3.15 to 3.17. Successful detection are shown in green and false alarms are shown in red bounding rectangles. The image frames vary from those that contain a single person to crowded people. In most of the cases, the detector does a very good job in detecting people. False alarms are usually

triggered on structures that have dominant vertical edges resembling the limbs of humans, *e.g.*, in figures 3.15(b), 3.16(c), 3.17(d), and 3.17(f). It can also be observed from figure 3.16(d), the detector is sensitive to pose variations due to either limb articulations or bent torso as it fails to detect the children. Another pitfall is the case of partial occlusion of legs; the detector is sensitive to occlusion of even small portion of legs, figures 3.15(c) and (d), 3.16(e), and 3.17(e).

Compared to the state-of-the-art, the detector trained on the INRIA dataset reflects an average performance that situates in between the best and least ranked detector performances. But, in terms of computation time, it exhibits the 3^{rd} best frame rate lagging behind **ChnFeats** and **LatSvm-V2**. The 0.53 fps achieved by this trained model runs short of real time requirements. On the other hand, the detector trained on the Ladybug dataset is, relatively speaking, simpler compared to the INRIA counter part. As a result it achieves a minimum of 1.66 fps (a maximum of 2.32 fps by altering training parameters). This is an added advantage as a majority of the methods in the state-of-the-art do not have the ability to automatically change the complexity of the trained detector based on the dataset; examples include Dalal and Triggs HOG and **HogLbp** which have fixed size feature vector irrespective of dataset. Albeit the improved frame rate, this faster detector version still is not suitable for real time experiments. But further improvement can be achieved by putting an emphasis on implementation optimization. For example, the **FPDW** detector uses the underlying principles of **ChnFeats** and optimizes the detection process by approximating the features over scale space resulting in a fast multi-scale detector that achieves approximately 6.5 fps on 640×480 image frames, a drastic increase over the 1.18 fps achieved using **ChnFeats** directly. In our case, we managed to further improve the speed of our detector by making use of a GPU implementation. The final detector implemented for GPU achieves 12.5 fps with 640×480 image frames and 7.5 fps with the downsized *Ladybug2* image frames of 1200×386 pixels. This is a significant gain in frame rate, especially compared to conventional methods, and how this improves robotic real time perception of people in its vicinity during navigation will be shown in the second part of this thesis. In the next chapter, chapter 4, further investigations carried out with similar framework employing heterogeneous pool of features for further frame rate improvements will be presented.

3.10 Conclusions

In conclusion, a person detection framework that makes use of the proven discriminant HOG features in a cascade configuration has been presented. A new feature selection technique based on mathematical programming has been devised to select features with good detection performance and less computation time. The complete final learning system has been validated on a proprietary dataset acquired using *Ladybug2* camera, a sensor which is interesting but surprisingly marginally used in the robotics community–perhaps due to the time consumption with the associated high resolution images. The methodology is also quite suitable for conventional cameras as illustrated by the evaluations on public dataset. The major contributions presented in this chapter have been published in [Mekonnen 2013a]. In the next chapter, this framework will further be investigated employing heterogeneous pool of features with substantial discriminatory and computation time differences.

Chapter 4

Mining Heterogeneous Features for Improved Detection

Contents

4.1 Introduction . . . 64
4.2 Framework . . . 64
4.3 Features and Weak Classifiers . . . 65
4.3.1 Heterogeneous Feature Set . . . 65
4.3.2 Weak Classifiers . . . 68
4.3.3 Computation Time . . . 68
4.4 Nodal Strong Classifier Learning Schemes . . . 69
4.4.1 Pareto-Front and AdaBoost . . . 69
4.4.2 Binary Integer Optimization and AdaBoost . . . 70
4.4.3 AdaBoost with Random Feature Sampling . . . 70
4.4.4 Computation Time Weighted AdaBoost . . . 70
4.5 Cascade Detector Learning . . . 72
4.6 Experiments and Results . . . 72
4.6.1 Implementation Details and Validation . . . 73
4.6.2 Results . . . 74
4.7 Discussions . . . 82
4.8 Conclusions . . . 84

4.1 Introduction

In the previous chapter, a people detection framework based on HOG features and Binary Integer Programming has been presented. The framework led to a detector with comparable detection performance to Dalal and Triggs [Dalal 2005] detector but with much improved computation time. In this chapter, we further investigate the framework by considering heterogeneous pool of features that have considerable variation in terms of discrimination and computation time. We have considered five family of the most commonly used features—Haar like features [Viola 2004], Edge Orientation Histograms (EOH) [Gerónimo 2007], Local Binary Patterns (LBP) [Mu 2008], Histogram of Oriented Gradients (HOG) [Dalal 2005]—with a boosted cascade detector configuration [Viola 2004]. To clearly outline the pros and cons of the BIP based framework, we also investigate alternative detector learning modes based on: (1) the classical AdaBoost with random feature selection; (2) computation time weighted AdaBoost; and (3) Pareto-Front analysis and AdaBoost. Similar to the previous chapter, the different approaches are evaluated using the Ladybug, INRIA, and Caltech datasets (presented in section 2.3). The additional variants are considered here, and not in the previous chapter, due to the increased diversity in terms of discrimination and computational cost and we wanted to be thorough to mine any possibility for computationally cheap detector.

The chapter is organized as follows. It begins with a presentation of the framework in section 4.2. Then, the heterogeneous features and associated weak classifiers are presented in section 4.3. Sections 4.4 and 4.5 present the different detector training modes in detail. In section 4.6 the experiments carried out along with obtained results are presented. Finally, the chapter ends with an extensive discussion and concluding remarks in sections 4.7 and 4.8 respectively.

4.2 Framework

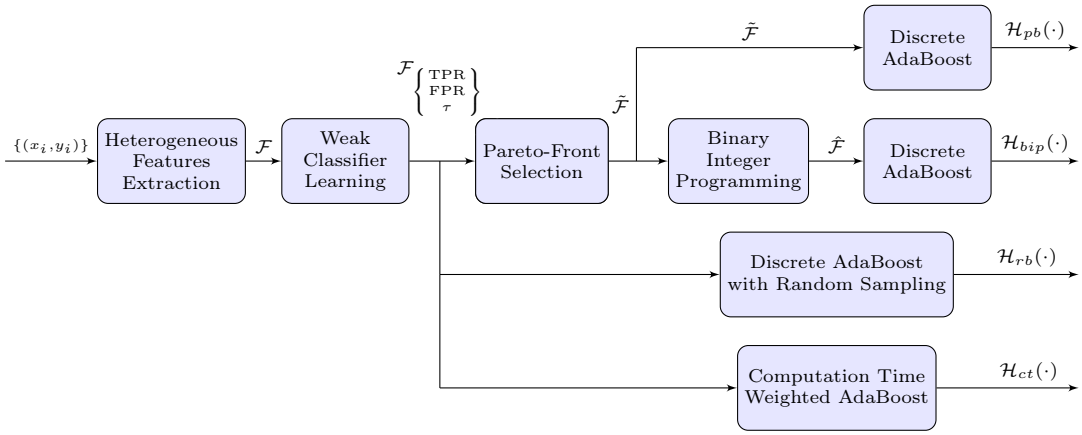


Figure 4.1: Investigated cascade node training schemes using heterogeneous pool of features.

Figure 4.1 visually summarizes the different approaches for detector training investigated in this chapter. Four different classifier learning schemes using heterogeneous features are represented in the framework. The block diagram shows the different strategies proposed to train a nodal strong classifier. For a cascade node, given labeled positive and negative training sets, the heterogeneous features, $\mathcal{F}$, described in section 4.3 are extracted and associated weak classifiers

trained. The first strategy (section 4.4.1) relies on the reduced feature set by Pareto-Front extraction, $\tilde{\mathcal{F}}$, and trains a nodal classifier on it using discrete AdaBoost. The second strategy is exactly the same framework based on BIP presented in chapter 3 applied on the heterogeneous feature pool, section 4.4.2. In the third (section 4.4.3), discrete AdaBoost is directly used to train a nodal strong classifier using a randomly sampled features from $\tilde{\mathcal{F}}$. Finally, the fourth strategy (section 4.4.4) proposes using a computation time weighted AdaBoost directly on $\mathcal{F}$ with random sampling. These proposed nodal classifier learning strategies are used to train corresponding cascade detectors.

4.3 Features and Weak Classifiers

This section presents the heterogeneous pool of features used for building an improved people detector along with the weak classifiers used for each family of features and computation time analysis carried out.

4.3.1 Heterogeneous Feature Set

In this work, we have chosen to use the following five family of features: Haar like features, Edge Orientation Histograms (EOH), Center-Symmetric Local Binary Patterns (CS-LBP), Color Self Similarity (CSS) features, and Histogram of Oriented Gradients (HOG). These choices are motivated mainly by two factors: (1) their frequent use in the literature for person detection, and (2) their complementary nature (in terms of both discrimination and computation requirements). EOH and HOG capture edge distributions, CSS focuses on color symmetry, and Haar-like and CS-LBP on intensity and texture variations.

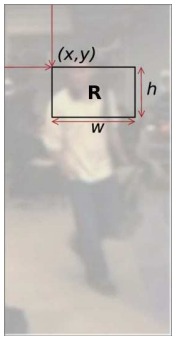


Figure 4.2: Feature region specification.

Each feature family is extracted within a given image window of 128×64 pixels denoted as R, a standard template size used prominently in people detection [Dollár 2012]. To generate the over-complete set of features, the position, width, and height of the region the features are computed is varied within the candidate window. In all references, (x, y) position refers to the top left corner of the region, relative to the top left corner of the candidate window, and (w, h) refers to the width and height of the region spanned for extraction, figure 4.2.

4.3.1.1 Haar Like Features

Haar like features represent a fast and simple way to compute region differences. These features have been extensively used for face, person, and various object detections, *e.g.*, [Papageorgiou 2000, Viola 2004, Lienhart 2002, Gerónimo 2007]. For a given feature, the response is obtained by subtracting the sum of pixels spanned by the black region from the sum of pixels spanned by the white region. To incorporate various measure, we have used the extended Haar like features from Viola and Jones [Viola 2004] and Lienhart and Maydt [Lienhart 2002], which contains upright and tilted filters of various configurations as shown in figure 4.3.

Let the operator $\Omega_{haar}(R, x, y, w, h, \varphi)$ denote the feature extracted (scalar value) in the overlaid region (x, y, w, h) with in the candidate window R using the Haar filter type φ. The over-complete set of Haar like pool of features, denoted as $\mathcal{F}_{haar}$, is obtained by extracting features for all x, y, w, φ combinations possible within R. In our implementation, a horizontal and vertical stride of 2 pixels are used to generate the over-complete set.

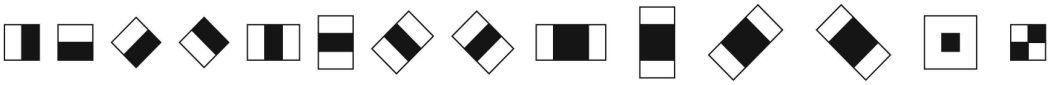


Figure 4.3: Set of extended Haar like feature types (configurations) used.

4.3.1.2 Edge Orientation Histogram (EOH)

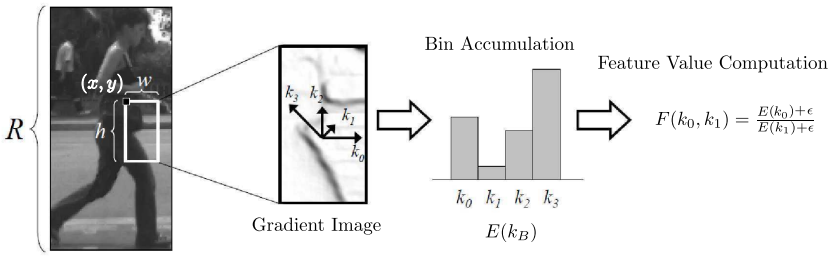


Figure 4.4: Illustration of a single EOH feature extraction from a given window (taken from [Gerónimo 2007]).

EOH is another popular feature set that has been used for people detection [Gerónimo 2007]. These features represent ratios of gradients computed from edge orientations histograms. Within a given overlaid region, first gradients are computed. Then, a gradient histogram is built by quantizing the gradient orientations. Finally, the ratios of each histogram bin with one another make up individual features. Similarly, let the operator $\Omega_{EOH}(R, x, y, w, h, k_0, k_1)$ denote the feature extracted in the overlaid region (x, y, w, h) within R by first building an edge orientation histogram and then taking the smoothed ratio of bins k_0 with that of k_1 as illustrated in figure 4.4. Consequently, the over-complete EOH feature pool set, denoted $\mathcal{F}_{EOH}$, is constructed by extracting feature values for all possible combinations of x, y, w, h, k_0, k_1 within R. In this work, gradient orientation quantization levels of 4 (shown to give best results in [Gerónimo 2007]) and horizontal and vertical strides of 4 pixels are used.

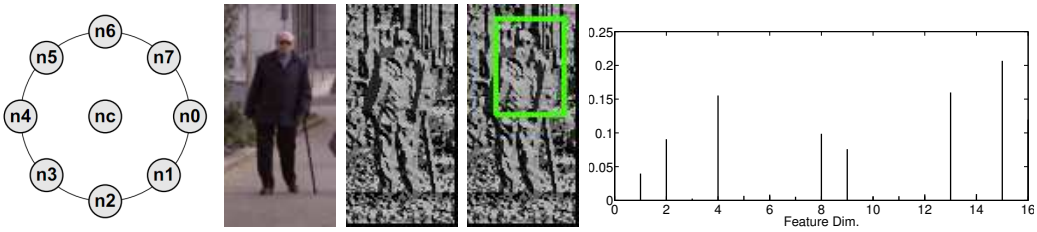
4.3.1.3 Local Binary Patterns (LBP)

Local Binary Patterns were initially proposed as a texture characterization features [Ojala 1996]. Since then, they have been used in many applications—primarily facial analysis, *e.g.*, [Zhao 2007], and person detection, *e.g.*, [Mu 2008]. To date, many variants of LBP have been proposed. In this work, we adhere to Center-Symmetric Local Binary Pattern (CS-LBP) variant owing to the short histograms it furnishes and demonstrated good results on person datasets [Heikkilä 2009].

$$CS - LBP = s(n0 - n4)2^0 + s(n1 - n5)2^1 + s(n2 - n6)2^2 + s(n3 - n7)2^3 \tag{4.1}$$

where, $s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases}$ and n0,...,n7 are gray scale pixel values (figure 4.5a).

In our implementation, CS-LBP is computed over a 3×3 pixel region (best results reported in [Heikkilä 2009]) by comparing the opposite pixels and adding a modulated term according to equation 4.1 with respect to figure 4.5a. This gives a scalar value less than 16 which is



(a) (b) (c) (d) (e)

Figure 4.5: CS-LBP feature extraction steps. (a) Pixel neighborhood for use with equation 4.1 (8-connectivity), (b) original candidate image, (c) dense CS-LBP per pixel computed values, (d) one specific feature specified by a bounding box, and (e) actual feature vector extracted from (d).

assigned to the center pixel. This is done for all the pixels in the window. A sample raw feature image is shown in figure 4.5c (the values are scaled to aid visibility). Finally, the actual feature vector is extracted by constructing a CS-LBP histogram (figure 4.5e) over a given overlaid region, figure 4.5d. Let, $\Omega_{CLBP}(R, x, y, w, h)$ denote the feature vector constructed by making histogram of all CS-LBP features within the region (x, y, w, h). Extracting feature vectors for all possible combinations of x, y, w, h within R with strides of 4 pixels in both direction gives the LBP feature pool, denoted $\mathcal{F}_{CLBP}$. The histograms have 16 bins corresponding to CS-LBP quantization levels.

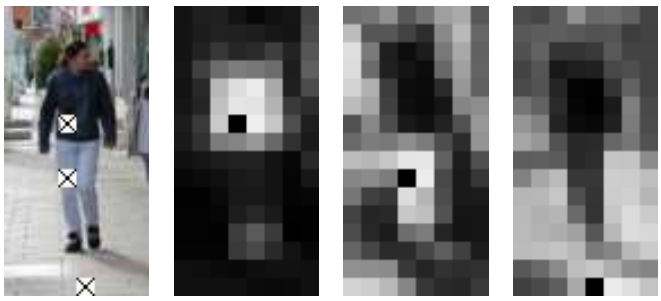


Figure 4.6: Illustration of sample CSS features.

4.3.1.4 Color Self Similarity (CSS)

Color features are rarely used in person detection because of the variability induced by clothing. Color, actually, shows local similarity even over clothing. CSS features, proposed by Walk *et al.* [Walk 2010], encode similarities in different sub-regions. To compute the features, first the image window is subdivided into non-overlapping blocks of 8×8 pixels and within each block a $3 \times 3 \times 3$ color histogram in HSV space is built with interpolation. Then, similarities are computed by intersecting individual histograms. In [Walk 2010], all histogram intersections values are used to define one feature vector. But, here, we define the intersection of one histogram block with the rest of the blocks as a single feature. With an 8×8 block size and 128×64 window size, there are 128 different blocks. The intersection of one block with the rest gives 127 scalar values (excluding intersection with itself). These scalar values all together make-up the feature vector computed for that specific block location. This is repeated for each block resulting in 128 different features, the CSS feature pool set ($\mathcal{F}_{CSS}$), of 127 dimensions each. Figure 4.6 shows three sample features

computed at the crossed block positions; observe how neighboring blocks show similarity.

4.3.1.5 Histogram of Oriented Gradients (HOGs)

These feature families have already been thoroughly presented in section 3.3. Briefly, we designate the operator $\Omega_{HOG}(R, x, y, w, h)$ that concatenates all feature blocks within the region, (x, y, w, h), as a single feature. The HOG feature pool set, $\mathcal{F}_{HOG}$, is obtained by extracting features using all possible values of x, y, w, and h in R—a total of 3360 variable dimensional features. The feature with the smallest length represents a single HOG block, *i.e.*, 36 dimensions, while the longest is made up of all block features concatenated with 3780 dimension, exactly the descriptor used by Dalal and Triggs.

Finally, the complete heterogeneous feature pool is determined by merging all heterogeneous feature pool sets, *i.e.*, $\mathcal{F} = \{\mathcal{F}_{Haar}, \mathcal{F}_{EOH}, \mathcal{F}_{CLBP}, \mathcal{F}_{CSS}, \mathcal{F}_{HOG}\}$. In consecutive sections, each individual feature is indexed by j, where $j \in \{1, 2, ..., |\mathcal{F}|\}$.

4.3.2 Weak Classifiers

The complete heterogeneous pool of features comprises of scalar and multi-dimensional features. For all scalar features, *i.e.*, Haar-like and EOH features, we have chosen to use decision trees as a weak classifier. A decision tree over a real valued scalar feature is equivalent to having multiple threshold values assigning different bands of the range for positive and negative samples. On the contrary, the considered CS-LBP, CSS, and HOG features are all multi-dimensional. In chapter 3, two weak classifiers suitable for multi-dimensional feature vectors have been discussed. In light of the detection performances exhibited, linear SVM is used as weak classifier for HOG and CSS feature vectors. Unlike HOG and CSS, the total number of CS-LBP features is quite high (table 4.1). Consequently, we have resorted to using Fisher's Linear Discriminant Analysis (LDA) [Fisher 1936] with decision tree as a weak classifier because of its comparatively short training duration. Given the large number of CS-LBP features, employing SVM would lead to an overwhelming training period. Each weak classifier, associated with a unique feature, is denoted as $\mathfrak{h}_j$ and maps each instance of the training set to a discrete label, $\mathfrak{h}_j : X \rightarrow \{-1, +1\}$.

4.3.3 Computation Time

The computation time of each feature is determined irrespective of any implementation optimization that can be done during detection, *e.g.*, use of caches to buffer some features. This helps establish an upper bound on it. For each feature considered, the computation time is made up of two components. A part associated with image pre-processing (including rudimentary feature preparation) that is mostly shared by features of the same family, and a second part pertaining to the feature extraction and necessary computation during detection (*e.g.*, multi-dimensional feature projection). For a feature indexed by j, these are represented as $\tau_{p,j}$ and $\tau_{e,j}$ consecutively; the combined computation time of that feature becomes $\tau_j = \tau_{p,j} + \tau_{e,j}$. These values are determined by averaging over $1,000$ times repeated iterations. The computationally cheapest feature, a two boxed horizontal Haar filter which takes $0.0535\ \mu s$ to compute on a core *i*7 machine, is used as a reference to report computation time for other features. The range of computation time for each feature family is reported in table 4.1.

Table 4.1 summarizes the characteristics of the heterogeneous pool of features considered. The total number of features in each family, the minimum and maximum feature computation time (both pre-processing, τ_p, and extraction, τ_e) along with the weak classifier used are listed.

Table 4.1: Feature pool summary with minimum and maximum feature computation time in each feature family. Time is reported as a multiple of the cheapest feature total computation time of $\mathfrak{u} = 0.0535\ \mu s$[a].

Feature Type	No of features	τ_{min}		τ_{max}		Weak Classifier
		$(\tau_p)_{min}$	$(\tau_e)_{min}$	$(\tau_p)_{max}$	$(\tau_e)_{max}$	
Haar like	672,406	0.6$\mathfrak{u}$	0.40$\mathfrak{u}$	1.88$\mathfrak{u}$	1.60$\mathfrak{u}$	Decision Tree
EOH	712,960	2.72$\mathfrak{u}$	2.11$\mathfrak{u}$	315.65$\mathfrak{u}$	2.10$\mathfrak{u}$	Decision Tree
CS-LBP	59,520	1.24$\mathfrak{u}$	14.26$\mathfrak{u}$	111.60$\mathfrak{u}$	282.04$\mathfrak{u}$	LDA + Decision Tree
CSS	128	560.75$\mathfrak{u}$	457.19$\mathfrak{u}$	560.75$\mathfrak{u}$	457.19$\mathfrak{u}$	SVM
HOG	3,360	10.59$\mathfrak{u}$	479.12$\mathfrak{u}$	315.75$\mathfrak{u}$	51103.80$\mathfrak{u}$	SVM

[a]Computed on a core i7 machine running at $2.4Ghz$

4.4 Nodal Strong Classifier Learning Schemes

As depicted in the block diagram in figure 4.1, four different techniques are considered to train a people detector based on heterogeneous pool of features. Each of these modes are discussed below in this section.

4.4.1 Pareto-Front and AdaBoost

The first detector training mode is based on Pareto-Front analysis and discrete AdaBoost. In this mode, a cascade node is trained initially using Pareto-Front extraction as a feature selection scheme, retaining only non-dominanted features with respect to MR, FPR, and computation time. This step results in the reduced set $\tilde{\mathcal{F}}$ from $\mathcal{F}$. Then, it uses discrete AdaBoost to build a strong nodal classifier, $\mathcal{H}_{pb}(\cdot)$. The same algorithm presented in algorithm 3.1 is used to extract the Pareto-optimal set from $\mathcal{F}$.

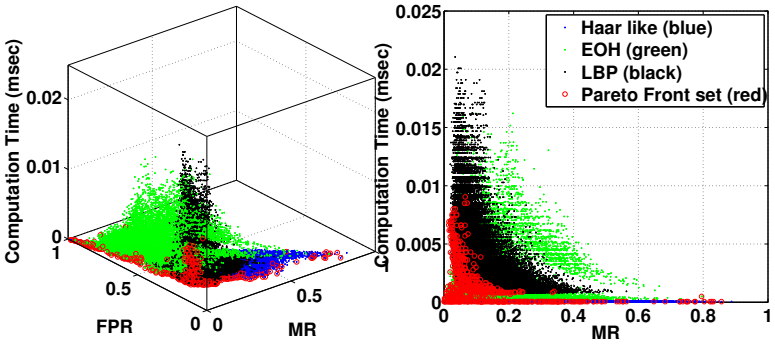


Figure 4.7: Sample Pareto-Front extraction with heterogeneous features. (Best viewed in color.)

Figure 4.7 illustrates a computed Pareto-Front, in 3D space and projected (MR vs computation time) 2D plot. To ease visibility only sub-sampled Haar like (shown in blue), EOH (in green), CS-LBP (in black) features are considered[1]. The determined Pareto optimal set is shown

[1]This is done for demonstration purposes but in the actual cascade construction all five feature pools without sub-sampling are considered.

with red circles. In the actual cascade configuration, the exact number of features extracted depends on their properties (MR, FPR, and computation time), but in our experiments the retained features never exceeded $1,500$.

4.4.2 Binary Integer Optimization and AdaBoost

This mode is exactly the same nodal classifier training presented in section 3.6. The only difference is, instead of using homogeneous HOG feature set, it uses the heterogeneous feature set discussed in this chapter. Similar to the approach in chapter 3, BIP is used to select the decisive feature set $\hat{\mathcal{F}}$ from $\tilde{\mathcal{F}}$ which is used to build the strong nodal classifier with discrete AdaBoost, $\mathcal{H}_{bip}(\cdot)$.

4.4.3 AdaBoost with Random Feature Sampling

Using AdaBoost directly on the extracted feature set, without any form of feature redaction, is the most classical detector learning scheme. It has been applied by various researchers following the pioneering work of Viola and Jones [Viola 2004]. For a given cascade node, the node trains a nodal strong classifier using AdaBoost iteratively. On each iteration, AdaBoost selects a single feature that minimizes the weighted error over the training samples, assigns a proper weight to the associated weak classifier, and adds it to the ensemble. Referring to algorithm 3.2, this entails evaluating the error term, ϵ_j, for all the weak classifiers in the pool. Given that we have a total of 1,448,374 weak classifiers in our pool, exhaustive search is not feasible. Therefore, we randomly sample a total number of $\mathcal{R}_s$ weak classifiers (in accordance with the relative proportion of features from each family) on each AdaBoost iteration and select the one that minimizes the classification error. The nodal strong classifier trained with this scheme is denoted as $\mathcal{H}_{rb}(\cdot)$.

4.4.4 Computation Time Weighted AdaBoost

This scheme is quite similar to the above discussed variant (section 4.4.3) based on random sampling and discrete AdaBoost. Here, we use a modified discrete AdaBoost that not only takes detection performance (*i.e.*, minimize ϵ_j) for selecting the best feature, but rather selects the one that minimizes a multiplicative term composed of feature computation time and detection error.

Similar to [Jourdheuil 2012], to use feature computation time, detailed in table 4.1, as a weighting term, we propose a smoothed normalized feature computation time, $\tilde{\tau}_j$, according to equation 4.2. $\beta \in [0, 1]$ is an exponential smoothing coefficient. $\tau_{max,l}$ denotes the maximum computation time registered within each distinct feature pool family, *i.e.*, $l \in \{Haar, EOH, CSLBP, CSS, HOG\}$.

$$\tilde{\tau}_j = \frac{\tau_j^{\beta}}{\sum_l \tau_{max,l}^{\beta}} \tag{4.2}$$

The computation time associated with each feature, $\tau_j = \tau_{p,j} + \tau_{e,j}$, is not constant (consequentially $\tilde{\tau}_j$ changes too). The exact value evolves during the classifier learning stage. It changes in two cases. The first is when a feature that has already been selected is considered in future cascade nodes, and the second is when a feature from the same family gets selected. In the prior case, the computation time of the selected feature is replaced by a constant time, τ_0, in future references which accounts for only memory access. In the latter case, the computation time for all of the features in the same family gets affected, specifically, the time associated with the pre-processing stage, $\tau_{p,j}$, is set to zero for all the features in that family. This is logical

and is done to favor features of the same family. For example, if a Haar feature is selected, it will be better to consider another Haar feature so the integral image computation can be done once for the area spanned by the two features, rather than considering another feature from a different family which will require a different pre-processing step. This way the computation time of the features within the same family will be levied contributing to speed up. Accordingly, the normalized computation time of all affected features is updated.

Recall that the original discrete adaBoost algorithm constructs a strong classifier by iteratively selecting the best weak classifier, $\mathfrak{h}_t$, based on the error distribution on the training set, ϵ_j, weights it, with α_t, and adds it to the ensemble. Each subsequent addition tries to correct the errors made by previously added weak classifiers. The modification here is to select the best weak classifier that minimizes the error weighted with a normalized computation time of the features, equation 4.3. This modification enables AdaBoost to select the feature (weak learner) that offers a compromise between computation time and detection error. This is detailed in algorithm 4.1 (main modifications on the classical one are shown in bold typeface). Again, due to the huge number of features, an exhausting search is not feasible and hence $\mathcal{R}_s$ randomly sampled features are used, as in section 4.4.3. The nodal strong classifier trained with this scheme is referred as $\mathcal{H}_{ct}(\cdot)$.

$$h_t = \arg\min_{h_j \in \mathcal{F}} \tilde{\tau}_j * \epsilon_j \tag{4.3}$$

Algorithm 4.1 Computation Time Weighted AdaBoost

procedure TRAIN_ADABOOST($\mathcal{F}$, $\{(x_i, y_i)\}_{i \in N}$)
  Initialize: $D_1(i) = \dfrac{1}{(n_+ + n_-)}$
  for t = 1,2,...,T **do**
    · Find the best weak learner $\mathfrak{h}_t$:
      → $\mathcal{F}_r \leftarrow$ randomly sample $\mathcal{R}_s$ features from $\mathcal{F}$
      → **Compute** $\tilde{\boldsymbol{\tau}}_{\boldsymbol{j}} = \dfrac{\boldsymbol{\tau}_{\boldsymbol{j}}^{\boldsymbol{\beta}}}{\sum_{\boldsymbol{i}} \boldsymbol{\tau}_{\boldsymbol{max,i}}^{\boldsymbol{\beta}}}$
      → $\mathfrak{h}_t = \arg\min_{\mathfrak{h}_j \in \mathcal{F}_r} \tilde{\tau}_j * \epsilon_j$ where $\epsilon_j = \sum_{i=1}^{n_+ + n_-} D_t(i)[y_i \neq \mathfrak{h}_j(x_i)]$
    · Compute weak learner weight: $\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$
    · Update data weight distribution: $D_{t+1}(i) = \dfrac{D_t(i) \exp(-\alpha_t y_i \mathfrak{h}_t(x_i))}{Z_t}$
  end for
end procedure

The computation time, ζ_k, of a trained cascade node k is determined straightforward by adding the computation time of each selected component features (associated weak learners), *i.e.*, $\zeta_k = \sum_{t=1}^{T} \tau_t$. The index t is used to signify reference of a selected feature and T represents the total number of features in this cascade node.

4.5 Cascade Detector Learning

A similar cascade detector learning scheme as the one presented in section 3.7 is adopted here. Each node k of the cascade is trained to fulfill a stipulated nodal TPR_k and FPR_k constraints. As discussed in section 4.4 above, we investigate four different learning schemes. Each scheme has its own unique way of tackling the problem. All in all, four different cascade detectors are developed using each nodal classifier learning scheme: a cascade using Pareto-Front with AdaBoost, with nodal classifiers $\mathcal{H}_{pb}(\cdot)$, referred as **Pareto+AdaBoost** henceforth; a cascade using BIP and AdaBoost, with nodal classifiers $\mathcal{H}_{bip}(\cdot)$, referred as **BIP+AdaBoost** henceforth; a cascade using AdaBoost with randomly sampled features, nodal classifiers $\mathcal{H}_{rb}(\cdot)$, referred as **Random+AdaBoost** henceforth; and, a cascade using the modified computation time weighted AdaBoost on a randomly sampled set of features, nodal classifier $\mathcal{H}_{ct}(\cdot)$, referred as **CTWeightedAdaBoost** henceforth.

4.6 Experiments and Results

In this section the different experiments carried out to investigate the performance of the proposed framework and obtained results along with commentaries are presented. In this chapter, principally, four different cascade detector training approaches, based on four differing nodal classifier learning schemes, using heterogeneous pool of features are presented. In this experimental section, we investigate which of these methods performs better taking both detection performance and computation time into consideration. In short, the experiments are focused on the following two aspects:

- Evaluation of feature selection and classifier learning strategy: The aim is to analyze the pros and cons of using each of the four different proposed classifier learning schemes. The four approaches are referred as **Pareto+AdaBoost**, **Random+AdaBoost**, **CTWeightedAdaBoost**, and **BIP+AdaBoost**.
- General comparative evaluation with the state-of-the-art: In this part, the performance of the best trained cascade detector (as it will become evident that of BIP+AdaBoost) is compared with the prominent approaches in the literature.

All evaluations are carried out on three datasets presented in section 2.3. Cascade classifiers are trained and tested on both the Ladybug and INRIA dataset. Then, the best performing cascade classifier trained on the INRIA dataset is tested on the Caltech dataset and compared with the state-of-the-art detectors.

Similar to chapter 3, we define the Average Speed Up (ASU)) criterion to compare the performance of a trained detector with respect to computation time. Recall that for a cascade detector the average computation time for a given candidate window is affected by the FPR of each node. Let K be the total number of nodes in the cascade, FPR_k be the false positive rate and ζ_k be the total computation time of the k^{th} node during detection. Assuming the nodal FPR characteristics hold on a generic input image, the average time spent on a test candidate window, ζ_{av}, can be estimated using equation 4.4.

$$\zeta_{av} = \sum_{k=1}^{K}(\prod_{z=0}^{k-1}\text{FPR}_z)\zeta_k \tag{4.4}$$

$$\text{ASU} = \frac{\zeta_{HOG}}{\zeta_{av}} \tag{4.5}$$

Using Dalal and Triggs [Dalal 2005] detector, which takes ζ_{HOG} per candidate window, as a reference, the **Average Speed Up (ASU)** over it is determined using equation 4.5. Consequently, the ASU values reported henceforth are with respect to Dalal and Triggs detector.

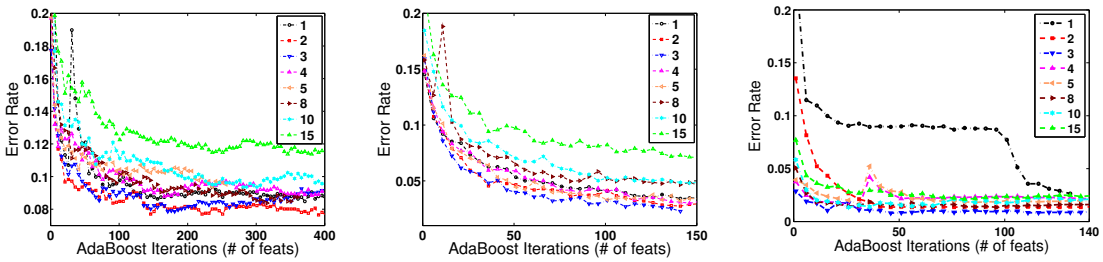
4.6.1 Implementation Details and Validation

The framework presented in this chapter relies on the framework presented in chapter 3. Consequently, from a software point of view, it relies on the same language using the same discussed open source libraries. The additions introduced in this chapter are custom developed.

Recall that the cascade node training is governed by two parameters: the nodal FPR and TPR. In all the experiments a nodal TPR value of 1.0 and FPR of 0.5 is used unless specified otherwise (the exception is the variant discussed in section 4.6.2.2). During training the TPR and FPR values exhibited by the weak classifiers or cascade node is determined on a separate validation set. Actually, the training dataset is initially divided into a 60% training and a 40% validation set; the new training set is used to train the weak classifiers and nodal classifiers whereas the validation set is used to determine detection performance exhibited. Once all feature selection is done, the node is retrained using the complete training set. This is applied when considering both the Ladybug and INRIA datasets (section 2.3).

In the following subsections, the validation steps taken to determine free parameters that are crucial for the performance of the different presented nodal classifier training schemes are discussed. These parameters are: (1) the depth of the decision trees used, (2) number of features randomly sampled ($\mathcal{R}_s$) with the Random+AdaBoost and CTWeightedAdaBoost strategies, and (3) the computation time smoothing coefficient (β) used in the CTWeightedAdaBoost variant. In all cases, a training and validation set from the Ladybug dataset is used.

4.6.1.1 Decision Tree Depth



(a) (b) (c)

Figure 4.8: Decision tree depth validation for (a) Haar like features, (b) EOH features, and (c) CS-LBP features.

The depth of the decision trees for Haar like features, EOH, and CS-LBP features (table 4.1) are validated as follows. Using only a single cascade node, a strong classifier is trained using AdaBoost with each individual feature families, and its performance on a validation set analyzed. Multiple runs are performed using different decision tree depths. On each AdaBoost iteration, only $2,000$ randomly sampled features are used to limit the validation time to a reasonable duration. For Haar like, EOH, and CS-LBP features, the error rate on the validation set as a function of the AdaBoost iteration for different decision tree depths is shown in figures 4.8a, 4.8b, and 4.8c respectively. Based on this, decision tree depths of 2, 3, and 3 are used for Haar like

features, EOH features, and LBP features respectively. The weak classifiers learned using these depths offer a better trade off between detection performance and over-fitting on the training set. Computing Fisher LDA weights, for CS-LBP features, per each node makes the classifier over-fit on the training set with deteriorated performance on the validation set. Hence, the Fisher LDA weights computed at the first node are used throughout the cascade by learning only new decision trees.

4.6.1.2 Number of Randomly Sampled Features ($\mathcal{R}_s$)

Both Random+AdaBoost and CTWeightedAdaBoost variants do not have a mechanism to reduce the entire feature set prior to the nodal classifier learning by AdaBoost. Given the vast number of features involved, looping through each feature set at each iteration of AdaBoost is infeasible. As is commonly done, *e.g.*, in [Zhu 2006, Wu 2008], at each AdaBoost iteration we use a randomly selected subset of features. According to Scholkopf and Smola [Scholkopf 2001](pp. 180), given set of samples, it can be guaranteed to sub sample amongst the best r_s percentage of estimates with a probability p by randomly sampling a sub sample of size $\frac{log(r_s)}{log(p)}$. This reduced set will do as well as considering the entire set with a probability p. In our case, to select amongst the best 5% features with a 99% probability, we need to sample a total of $\frac{log(0.05)}{log(0.99)} \approx 299$ samples. In our implementation we use 3000 features which is way above the suggested number of samples and guarantees to obtain the relevant features with a high probability. Hence, $\mathcal{R}_s = 3000$. Bear in mind, the sampling is actually done in proportion to the total number of features contributed from each feature family.

4.6.1.3 Computation Time Smoothing Coefficient (β)

Similarly, the exact value of β, the computation time smoothing exponential factor, to use in the computation time weighted AdaBoost is determined empirically through a validation step. The CTWeightedAdaBoost is used to learn a single nodal cascade using different β values on a subset of the training set. Then the classification errors on a validation set and the conglomerated computation time of the trained node is determined to select the best value that offers a good trade-off. Figure 4.9 shows the validation result plots for different values of β. Clearly, higher β reduces smoothing, in effect, features with low computation time dominate improving speed but with poor detection performance. Lower values favor complex features. As a compromise, a β value of 0.2 is used to train the final cascade classifier in this scheme.

4.6.2 Results

The results corresponding to all experiments are reported in this section categorized under each dataset.

4.6.2.1 Ladybug Dataset

With this dataset, four different cascade detectors are trained. In all cases, a nodal FPR of 0.5 and TPR of 1.0 is stipulated. The main results obtained are depicted in figure 4.10 and summarized in table 4.2. Clearly Pareto+AdaBoost results in the best detection performance, 2.9% MR, followed by Dalal and Triggs detector trained on this dataset, 3.0%, at 10^{-4} FPPW. CTWeightedAdaBoost shows the lowest detection performance with a 10% MR at 10^{-3} FPPW, but it manages to learn a detectors that is 1.8× faster than Dalal and Triggs HOG. In terms of detection, BIP+AdaBoost trails behind Random+AdaBoost with marginal loss. But, the most important result to notice is that BIP+AdaBoost results in a drastic 42.7x speed up over Dalal

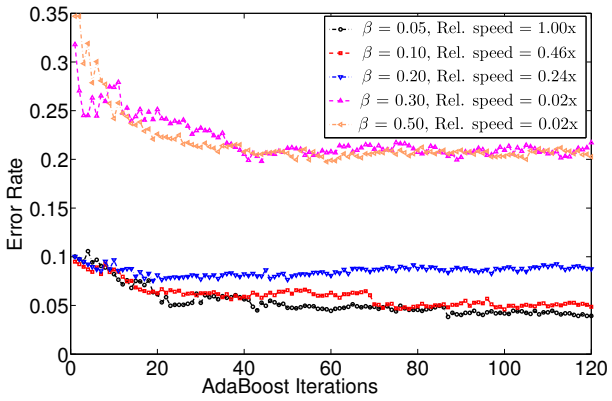


Figure 4.9: Error rate on a validation set using the computation time weighted AdaBoost trained with different β values.

and Triggs with only a 7% loss in MR at 10^{-4} FPPW. The main reason for this speed up is that BIP+AdaBoost systematically uses cheap features in the initial stages of the cascade and only starts using computationally expensive features at later stages. The trained classifier has 10 cascade nodes with CSS features initially appearing at the 6^{th} node and HOG at the final stage; figure 4.12 depicts this showing the selected features at some of the nodes.

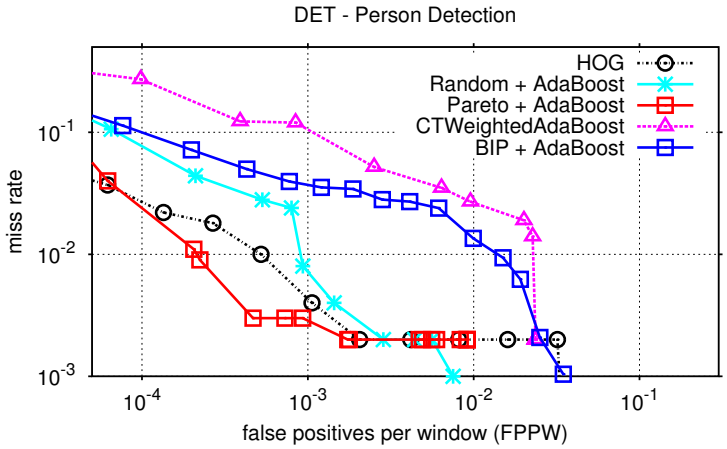


Figure 4.10: DET of different detectors trained and tested on the Ladybug dataset.

Apparently, Pareto+AdaBoost and Random+AdaBoost result in worsened speeds. This is because AdaBoost always privileges the most discriminant feature, irrespective of computation cost, from the pool of features passed to it, and both Pareto-Front extraction and random sampling are likely to pass such kind of complex features. Consequently, the set of features selected in the first node result in a conglomerate that is effectively computationally demanding than Dalal and Triggs detector. CTWeightedAdaBoost improves upon this by selecting a slightly less complex feature. Figure 4.11 shows the features selected in the initial cascade of all detectors trained. All three, Random+AdaBoost, Pareto+AdaBoost, and CTWeightedAdaBoost, have HOG features in this node contributing to reduced speed; on the contrary, for BIP+AdaBoost,

Table 4.2: Summary of the cascade detector trained on the Ladybug dataset. Miss Rate is reported at 10^{-4} FPPW.

Detector	Feature Proportion					MR	ASU
	Haar	CS-LBP	CSS	EOH	HOG		
[Dalal 2005]	–	–	–	–	100%	3.0%	1.0x
Pareto + AdaBoost	10.7%	0.0%	0.0%	0.0%	83.7%	2.9%	0.7x
CTWeightedAdaBoost	53.3%	33.3%	0.0%	10.0%	3.3%	25.0%	1.8x
Random + AdaBoost	51.6%	6.2%	1.5%	36.0%	4.7%	8.0%	0.6x
BIP + AdaBoost	54.3%	8.6%	8.5%	25.7%	2.8%	10.0%	42.7x

only CS-LBP and Haar features are used. These result are obtained using a fixed nodal FPR of 0.5 for all constructed nodes and the obtained results are very precise that altering the FPR is not necessary.

The proportion of features present from each family is also consistent with the underlying training scheme adopted (table 4.3). Both CTWeightedAdaBoost and BIP+AdaBoost emphasize on computation time, accordingly, they have the highest proportion of Haar features in their trained model. They also have the least proportion of HOG features taking only 3.3% and 2.8%, respectively, of the total proportion of features. Pareto+AdaBoost favors complex features with superior detection performance which explains the 83.7% HOG features presence.

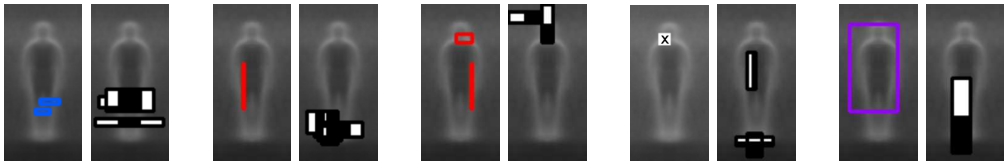
Compared to the results obtained using HOG features and BIP in table 3.3, the result obtained using heterogeneous features with BIP (BIP+AdaBoost) is superior in terms of computation time. But, in terms of detection performance, the HOG features and BIP variants trained with a nodal FPR of 0.4 show better performance at 10^{-4}, though at higher FPPWs they show poor performance relative to the Pareto+AdaBoost and Random+AdaBoost variants.

(a) Random+AdaBoost (b) Pareto+AdaBoost (c) CTWeightedAdaBoost (d) BIP+AdaBoost

Figure 4.11: The features selected and used in the first node of the cascade under the different learning approaches with the Ladybug dataset superimposed on an average human gradient image. Black and white rectangular regions show Haar features, blue for CS-LBP, crossed white boxes represent CSS features and their position indicates the reference block, and finally, violet shows the spatial region spanned by the concatenated HOG blocks.

4.6.2.2 INRIA Dataset

Similar results obtained for the INRIA dataset are shown in figure 4.13 and summarized in table 4.3. As this dataset is challenging, two variants of the BIP+AdaBoost classifier are trained. In the first case, a fixed nodal FPR of 0.5 is used for all nodes, called **BIP+AdaBoost(Fix)**. In the second case, an adaptive FPR is employed which starts at 0.3 in the initial stage and continues training nodes, whenever a solution for the BIP optimization does not exist, this



(a) node-0 (b) node-1 (c) node-2 (d) node-5 (e) node-9

Figure 4.12: Illustration of the different features selected on different nodes of cascade (superimposed on an average human gradient image) of BIP+AdaBoost trained on Ladybug dataset. Black and white rectangular regions are Haar features, red for EOH, blue for CS-LBP, crossed white boxes for CSS, violet HOG features.

constraint is relaxed/incremented by 0.1 and the procedure continues from that node likewise until all negative samples are depleted. This is called **BIP+AdaBoost(Ad)**. Again, the best detection results at 10^{-4} FPPW are obtained by the Random+AdaBoost and Pareto+AdaBoost variants. But, this time both variants of BIP+AdaBoost beat Dalal and Triggs detector at 10^{-4} by more than 2%. On top of this, the BIP+AdaBoost(Fix) achieves a 15.6x speed up while that of BIP+AdaBoost(Ad) trails with a 9.22x speed up. Random+AdaBoost, Pareto+AdaBoost, and CTWeightedAdaBoost variants on the other hand result in increased computation time (even with respect to [Dalal 2005]). The reason is all these three start off with complex features in the initial nodes. Figure 4.14 shows the features selected in the first node of all trained detector variants. Only the ones employing BIP do not have HOG features in the initial stage. Even though the CTWeightedAdaBoost variant does not result in a significant boost in speed, it is twice as much faster as its random counterpart (Random+AdaBoost) with marginal loss in miss rate. The explicit computation time consideration does help even in this case.

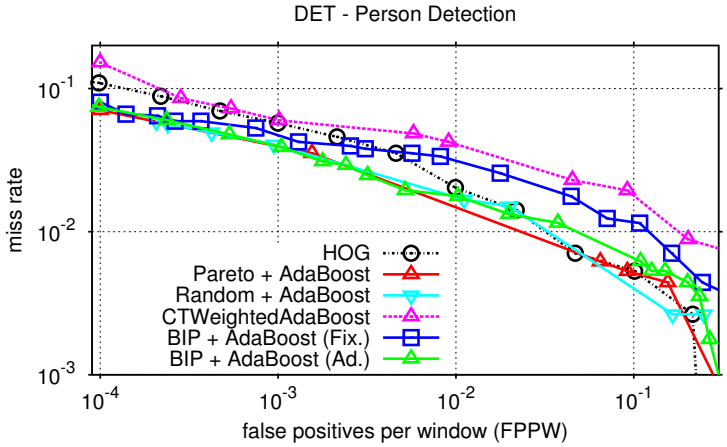


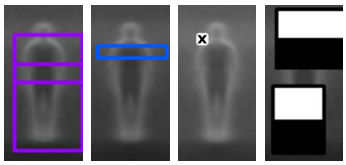
Figure 4.13: DET of different detectors trained and tested on the INRIA dataset.

Concerning the BIP+AdaBoost variants, as the initial FPR constraints are stringent on the BIP+AdaBoost(Ad) variant, it favors relatively discriminative features with increased computation time. This contributes to its superior detection performance, over BIP+AdaBoost(Fix), throughout the FPPW range shown in figure 4.13. Observe in table 4.3, there are more proportion of Haar like features (5.4% more) and less proportions of HOG features (2.0% less) in the fixed variant compared to the adaptive variant which results in the increased speed.

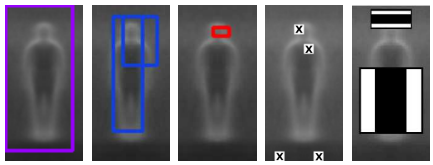
Table 4.3: Summary of the cascade detector trained on the INRIA datasets. Miss Rate is reported at 10^{-4} FPPW.

Detector	Feature Proportion					MR	ASU
	Haar	CS-LBP	CSS	EOH	HOG		
[Dalal 2005]	–	–	–	–	100%	11.0%	1.0x
Pareto + AdaBoost	42.8%	14.5%	7.8%	25.6%	9.3%	7.0%	0.4x
Random + AdaBoost	26.3%	10.8%	3.7%	53.5%	5.6%	6.0%	0.4x
CTWeightedAdaBoost	86.7%	9.1%	2.4%	0.0%	3.9%	14.6%	0.8x
BIP + AdaBoost(Fix)	60.4%	10.8%	8.0%	9.7%	11.0%	8.0%	15.6x
BIP + AdaBoost(Ad)	55.0%	14.6%	8.1%	9.3%	13.0%	7.4%	9.22x

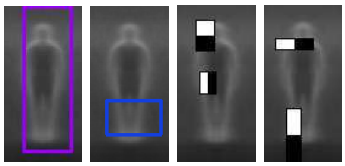
Figure 4.16 shows histogram of the selected features, with relative proportions, for the first 9 nodes of both the fixed and adaptive variants. Clearly, the fixed variant initially uses cheaper features and increases along the cascade both in number and complexity. On the contrary, for the variable variant, complex features appear in the initial nodes and increase in number along the cascade. Figure 4.15 illustrates a few of the selected features overlaid on an average human gradient image for BIP+AdaBoost(Ad). Observe that all selected features capture discriminant facets of people.



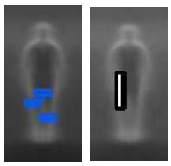
(a) Random+AdaBoost



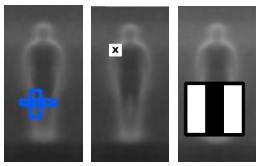
(b) Pareto+AdaBoost



(c) CTWeightedAdaBoost



(d) BIP+AdaBoost(Fix)

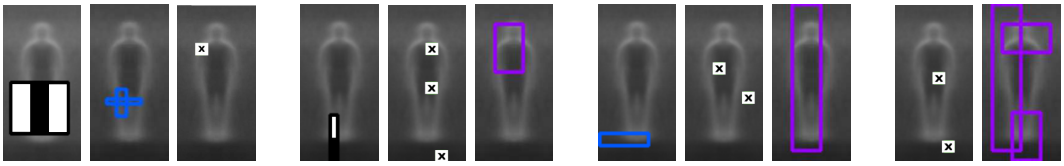


(e) BIP+AdaBoost(Ad)

Figure 4.14: The features selected and used in the first node of the cascade trained on the INRIA dataset superimposed on an average human gradient image. Black and white rectangular regions are Haar features, blue for CS-LBP, red for EOH, crossed white boxes for CSS, violet HOG features.

The BIP+AdaBoost detector variants trained with heterogeneous pool of features show better performance both in terms of detection as well as computation time compared to the detector trained using only HOG features, table 3.4. For example, the detector trained under the same FPR conditions, BIP+AdaBoost(Fix), achieves a 4.3% improved miss rate at 10^{-4} FPPW with a $\approx 7.0\times$ faster detection speed compared to the HOG only with BIP variant. This attests the complementary nature of the heterogeneous features–especially, taking both discrimination and speed criteria.

Figure 4.17 shows the comparative full image evaluation on the INRIA full image test set. Please refer to section 2.2.4 for an explanation of the other state-of-the-art approaches. On



(a) node 0 (b) node 1 (c) node 2 (d) node 8

Figure 4.15: Sample depictions of the heterogeneous features selected at different nodes of the cascade BIP+AdaBoost(Ad) trained on the INRIA dataset using an adaptive FPR. Black and white rectangular regions are Haar features, blue for CS-LBP, crossed white boxes for CSS, violet HOG features.

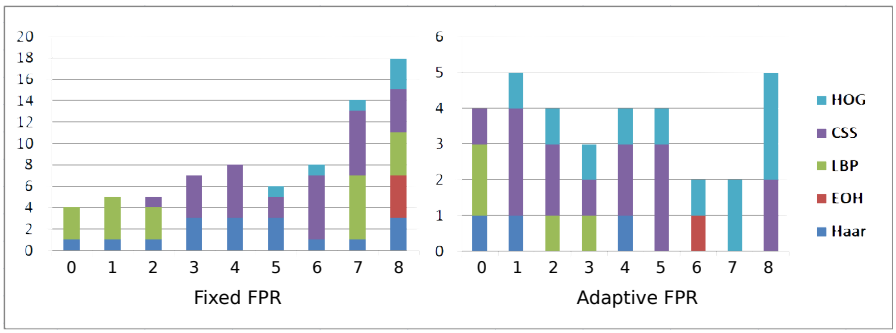


Figure 4.16: Histogram of selected features in the first 9 nodes of the model trained on the INRIA dataset using both fixed FPR of 0.5 and adaptive FPR.

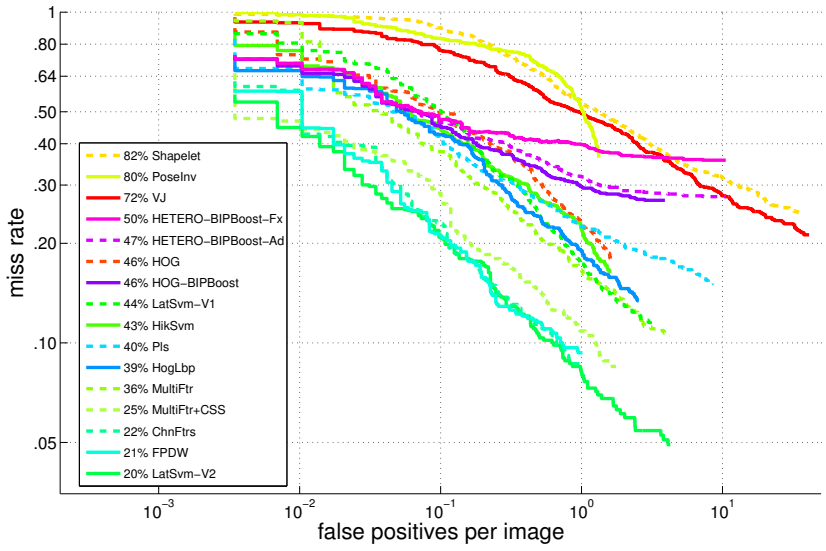


Figure 4.17: Comparative full image evaluation on the INRIA test set. Refer to section 2.2.4 for description of the listed detectors.

the figure, both BIP+AdaBoost variants presented in this chapter are labeled as **HETERO-BIPBoost-Fx** and **HETERO-BIPBoost-Ad** to dissociate them from the **HOG-BIPBoost** presented in chapter 3. To generate these results, a Pairwise Max non-maximal suppression [Dollár 2012] with an overlap threshold of 0.65 is used. The log-average miss rate of these detectors is very comparable with Dalal and Triggs HOG. In fact at lower FPPI values the proposed BIP based detectors exceed Dalal and Triggs HOG by exhibiting more than 10% reduction in miss rate. For all FPPI values less than 0.1 the BIP variant consistently supersedes Dalal and Triggs HOG. Clearly the BIP+AdaBoost variant with adaptive FPR shows better detection performance than the fixed FPR variant with a 3% reduced log-average miss rate. The HOG-BIPBoost version shows a marginally improved detection, owing to its constituent more discriminative features, *i.e.*, HOG, over the Heterogeneous counterparts.

Speed wise, using the computation speed reported in [Dollár 2012] for people more than 100 pixels in a 640×480 image, our detectors achieves 2.3 frames per second (fps) for the adaptive variant, and 3.9 fps for the fixed FPR variant trained on the INRIA dataset. These values are amongst the top best only exceeded by **FPDW** which achieves approximately 6.5 fps. But as mentioned previously, **FPDW** uses the underlying principles of **ChnFeats** and optimizes the detection process by approximating the features over scale space. Similar techniques can be used to further improve the fps of our detector. On the other hand, the model trained on the Ladybug dataset, achieves 10.6 fps on the simpler dataset (for images of 640×480 size). This is an added advantage as a majority of the methods in the state-of-the-art do not have the ability to automatically change the complexity of the trained detector based on the dataset; examples include Dalal and Triggs HOG and **HogLbp** which have fixed size feature vector irrespective of dataset.

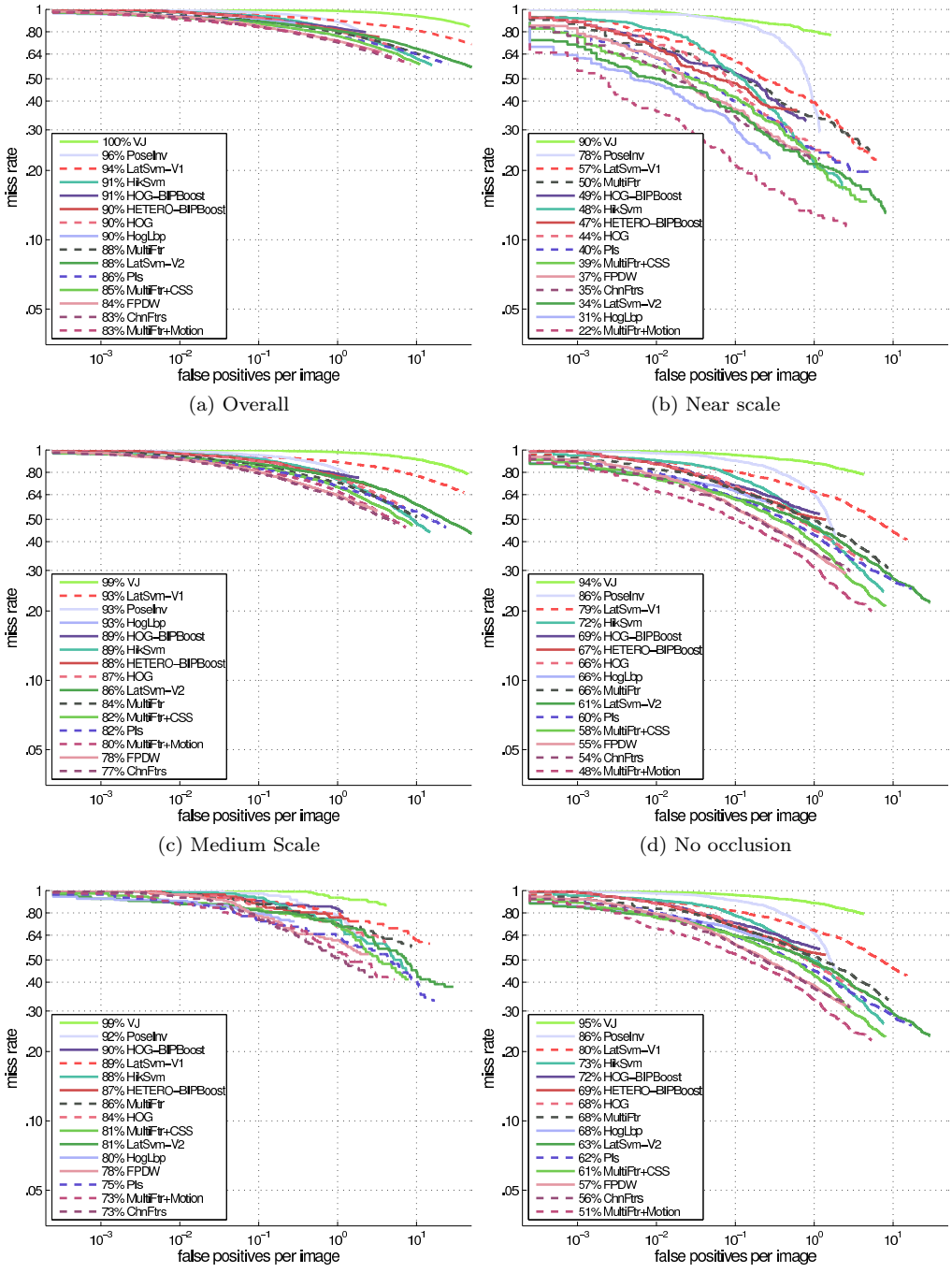
Table 4.4: Computation time comparison with the state-of-the-art. The values for the different detectors are taken from [Dollár 2012]. These values are determined on a 640×480 sized images detecting people with a minimum height of 100 pixels.

Detector	Shapelet	PoseInv	VJ	**HETERO-BIPBoost (Fix)**	HETERO-BIPBoost (Ad)	HOG	HOG-BIPBoost	LatSvm-V1	HikSvm	Pls	HogLbp	MultiFtr+CSS	ChnFtrs	LatSvm-V2
Fps[a]	0.05	0.47	0.45	**3.9**	2.3	0.24	0.53	0.4	0.19	0.02	0.06	0.03	1.18	0.63

[a]Run times of all detectors are normalized to the rate of a single modern machine [Dollár 2012].

4.6.2.3 Caltech Dataset

We evaluated the best BIP+AdaBoost variant, the one trained using adaptive FPR values, using the Caltech dataset. In the evaluations, this variant is termed as **HETERO-BIPBoost** to signify it is trained using heterogeneous features, with BIP and AdaBoost classifier learning. The full image evaluation results for the different categories are shown in figure 4.18. In the evaluations, the HOG-BIPBoost variant that is trained with only HOG features is also shown.



(a) Overall (b) Near scale

(c) Medium Scale (d) No occlusion

(e) Partial occlusion (f) Reasonable

Figure 4.18: Full image evaluation results on the Caltech test dataset.

In all cases, the HETERO-BIPBoost variant showed better detection performance than the HOG-BIPBoost variant, but it trailed behind Dalal and Triggs HOG. On the overall evaluation, the HETERO-BIPBoost actually showed a 90% log-average miss rate equal to Dalal and Triggs

HOG. The HETERO-BIPBoost like HOG-BIPBoost does well with near scale pedestrians (as the training INRIA dataset is composed mainly of people under near scale) with a 47% log-average miss rate. It also achieves 1% less log-average miss rate than Dalal and Triggs HOG on the reasonable set. Given these detection performance with the 9.22× frame rate improvement over Dalal and Triggs HOG makes HETERO-BIPBoost by far superior than Dalal and Triggs HOG. It also gives it a competitive edge in near scale and reasonable categories should there be a need for fast detection requirements.

4.7 Discussions

This chapter presents a people detection framework based on heterogeneous pool of features. The feature pool constitutes Haar like features, CS-LBP, EOH, CSS, and HOG features. These features capture varying cues relevant to people in an image. They are highly variable in terms of discriminative ability as well as computing time. For example, HOG features are the most discriminative ones and take relatively higher computation time than the rest, whereas on the contrary Haar like features tend to be the least discriminative ones and yet quite fast to compute. Given this kind of feature pool with diverse characteristics, extra care should be given to the way these features are mined to build a people detector. In general the two main possibilities are either to use all information, by concatenating individual features, to construct a very high dimensional single feature vector, or favor a sparser solution by employing some form of feature selection to reduce the set to a few performant ones. In our case, the first is out of the question given the huge number of features which would lead to a non-realistic classifier training, and hence all considerations are focused on the second alternative.

Settling on feature selection, to mine relevant features out of the pool, we have investigated four different approaches all based on the popular AdaBoost with cascade of nodes detector. The approaches are: Random+AdaBoost, which uses discrete AdaBoost to learn a strong nodal classifier by randomly sampling $\mathcal{R}_s$ features on each boosting iterations and adding the best one to the ensemble; Pareto+AdaBoost, which uses Pareto-Front analysis to retain only non-dominated features with respect to TPR, FPR, and computation time, and then builds a nodal strong classifier with discrete AdaBoost and the retained subset of features; CTWeightedAdaBoost, which makes use of a modified AdaBoost that selects the best features using the classification error weighted by a computation time measure to enable AdaBoost to give consideration for feature computing speed; and finally, BIP+AdaBoost which uses a discrete optimization formulation based on BIP to retain the minimum number of features that fulfill the stipulated detection performance with the minimum combined computation time. All four approaches have been evaluated on proprietary and public datasets thoroughly and compared to the state-of-the-art. In terms of detection, all four do well, for example achieving between 6.0% and 14.6% miss rate at 10^{-4} FPPW on the INRIA dataset which confirms that considering heterogeneous features is relevant.

The Random+AdaBoost variant is the most straight forward solution and it has been applied many times in the literature. The random sampling avoids the need to iterate through all features in the pool, at each boosting iteration, which would have otherwise made the training non-realistic. As long as the number of randomly sampled features is high, it will lead to comparable results as the exhaustive search. In terms of detection, this approach is expected to lead to the best (compared to the four variants) result as it always picks the best discriminative features iteratively. The downside of this approach is, reflecting the properties of AdaBoost, it always selects features based on classification performance blind to feature computation time. If there are two competing features with only a slight difference in classification error but big difference in

computation time, the minimum error feature will be selected. This characteristics is exemplified in the evaluations as it leads to the worst detector speed, 0.4× Dalal and Triggs HOG, with amongst the best detection performance, *e.g.*, 6.0% MR at 10^{-4} FPPW on the INRIA dataset, in all cases.

Taking the shortcomings of Random+AdaBoost, we propose and investigate CTWeightedAdaBoost variant. This variant is similar to Random+AdaBoost, but on each boosting iteration, AdaBoost selects the feature with the minimum weighted error, classification error down weighted by a normalized feature computation time. This actually gives a chance to cheap features that exhibit marginally less classification error compared to computationally intensive feature variants. In the experimental results, this is attested as it achieves approximately 2× and 3× as much faster detector compared to the Random+AdaBoost variant on the Ladybug and INRIA trained models respectively. But, the modification down plays its detection performance leading to reduced detection rate. For example it achieves a 5.6% MR reduction at 10^{-4} FPPW compared to Random+AdaBoost on the INRIA evaluation.

The third investigated scheme is the Pareto+AdaBoost variant. In this strategy, an initial feature selection is applied using Pareto-Front analysis by retaining dominant features with respect to TPR, FPR, and computation time. The main advantage of this approach is that it avoids the exhaustive search that needs to be done by AdaBoost and yet is guaranteed to pass on the most performant features. This is clearly seen by the low miss rate it exhibits on the test cases, 2.9% and 7.0% MR at 10^{-4} FPPW on the Ladybug and INRIA dataset evaluations respectively. But, again as long as computationally intensive discriminant features exist in the selection, which is actually the case as observed in the experiments, AdaBoost is bound to greedily favor the discriminant ones leading to computationally demanding detector. These remarks are seen on the evaluation results, for example on the INRIA test set, it achieves the second highest detection rate, 7.0% miss rate at 10^{-4} FPPW, with the least frame rate, more than twice slower than Dalal and Triggs HOG.

The fourth investigated scheme, BIP+AdaBoost, makes explicit optimization to select the features that achieve the required detection performance with the minimum possible computation time. The two modes investigated in this scheme are leaned using a fixed nodal FPR and an adaptive nodal FPR. Both variants result in a detector that is most considerate as both detection speed aspects are taking into account to come up with the best compromise. The BIP+AdaBoost(Fix) variant for example achieves a 10.0% and 8.0% MR at 10^{-4} on the Ladybug and INRIA datasets respectively. It contains significant proportions (more than 54% on the Ladybug model and more than 60% on the INRIA model) of Haar features and less proportions of the costly features, *e.g.*, only 2.8% and 11% HOG features in both datasets respectively. This helps it achieve a 42.7× and 15.6× speed up over Dalal and Triggs HOG using the Ladybug and INRIA trained models respectively. Its adaptive variant trained on the INRIA dataset improves the detection further achieving a 7.4% MR at 10^{-4} FPPW with a 9.22× speed up over Dalal and Triggs HOG. Consistently, its trained detector has more proportion of Haar features (55.0%) and less proportion (13.0%) of HOG features. Hence, it can be safely concluded that the BIP based detector variants are the most considerate ones as they work on both detection and speed aspects to come up with the best compromise. The 2.3 and 3.9 fps achieved by the adaptive and fixed variants are amongst the best in the state-of-the-art (table 4.4). If optimizations during detection, for example like feature value approximation over scale space like **FPDW**, are not considered, it is in the forefront of all compared state-of-the-art detectors with respect to computation time.

With the full image evaluation on the INRIA test set, both the fixed and adaptive variants achieve a 50.0% and 47% log-average miss rate trailing Dalal and Triggs HOG by only a 1% loss. Even with this marginal loss, the speed improvements would still make them superior

over it. Similarly, these exhibited results are also consistently observed on the Caltech dataset. The BIP+AdaBoost(Ad) which is referred as HETERO-BIPBoost in the Caltech evaluation results (figure 4.18) achieves an equivalent overall score to that of Dalal and Triggs HOG with a 90% log-average miss rate. Mind you this is only 7% less than the best results obtained by **MultiFtr+Motion**, which according to table 4.4 is slower by a factor of 76.7× compared to this BIP variant. In near scale, HETERO-BIPBoost achieves a 47% log-average miss rate, which even supersedes the HOG-BIPBoost detector by 2%. HETERO-BIPBoost, like the HOG-BIPBoost variant, suffers the most in the presence of partial occlusions and medium scaled people achieving an 87% log-average miss rate in both cases. From all of the above results, it can be observed that BIP+AdaBoost variants exhibit stable and consistent detection performance comparable with Dalal and Triggs HOG under different datasets and always results in a significant improvement over detection speed.

Another advantage of the BIP based framework is its flexibility with respect to computational resource constraints and detection requirement. On any dataset, the stipulated detection parameters used during training (nodal TPR and FPR) can either be made stringent or relaxed to learn a model that can either consume more or less computational resources respectively, giving explicit control on the detection vs speed trade off. This has been demonstrated with the adaptive and fixed detector variants trained on the INRIA dataset. By tightening the constraints (initially strict and eventually relaxed along the cascade pipeline), a 47% log-average miss rate is achieved, which is better than the 50% achieved using the fixed variant. But, these modifications led to a relative speed ratio of $\frac{15.6}{9.22} \approx 1.7$. Hence, depending on the application, these constraints could be modified accordingly, for example they could be relaxed until a tolerable detection loss is achieved to fulfill real time requirements.

Additionally, another flexibility of the framework is its ability to adapt the complexity of the learned model to the challenge inherently present in the training dataset. The framework takes profit of the underlying challenge manifested by the training dataset to furnish an appropriate detector model. For simpler datasets, it furnishes simpler model with increased frame rate, *e.g.*, the 10.6 fps achieved with the Ladybug dataset trained model contrary to the 3.9 fps achieved with the INRIA dataset. This quality would enable developing a detector that is suited for specific scene/domain that reflects on the detection challenge, for *e.g.*, for indoor open environment (like the hall of a shopping mall) application that might feature less background clutter with upright people having less pose variability, with faster frame rates. Most of the detectors listed in the state-of-the-art do not have the ability to automatically change the complexity of the detector based on the dataset. As an example, consider Dalal and Triggs HOG [Dalal 2005], **HogLbp**, **LatSvm-V1**, which have a fixed size high-dimensional classifier that is always fixed irrespective of the dataset.

4.8 Conclusions

In this chapter, different strategies to train a people detector using heterogeneous pool of features have been investigated. Various experiments have been carried out to investigate the advantages and shortcomings of each strategy using proprietary and multiple public datasets. The obtained results ascertain that complementary heterogeneous features lead to improved detection performance, and under explicit consideration of computation time, lead to improved frame rate as well. The different results also show the superiority of the BIP based feature selection strategy proposed in chapter 3. The proposed BIP strategy is quite capable in taking advantage of the diversity that exists in the feature pool from detection as well as speed perspectives. Further improved frame rates can also be achieved by parallelizing the trained model with the help of

specialized hardwares like a Graphical Processing Unit (GPU).

All in all, the BIP framework for feature selection has been carefully investigated and compared with the state-of-the-art in the previous (chapter 3) and current chapter. The various contributions made in this chapter have been published in [Mekonnen 2013e, Mekonnen 2013f, Mekonnen 2014a].

Part II

COOPERATIVE PERCEPTION FOR TRACKING PEOPLE

Chapter 5

Cooperative Perception and Multi-person Tracking: An Overview

Contents

5.1 Introduction 89
5.1.1 Environment Fixed Sensors 90
5.1.2 Mobile Sensors and Sensor Fusion Modes 90
5.1.3 Environment Fixed and Mobile Sensors 92
5.2 Multi-Person Tracking 92
5.2.1 Overview 92
5.2.2 Bayesian Formulation 95
5.2.3 MCMC- and RJMCMC-Particle Filters 96
5.2.4 Evaluation Metrics 99
5.3 Conclusion 100

5.1 Introduction

This chapter is intended to first provide a brief overview of the current trend in automated multi-person detection and tracking from the utilized system mode (configuration) perspective. Second, it presents the necessary foundations in the formulation of multi-person tracking (in section 5.2). The materials presented in this and the next section will help put the developments made in subsequent chapters more apparent.

Broadly speaking, the literature in automated multi-person detection and tracking encompasses works that use sensors fixed in the environment and those that use mobile sensors (either mounted on a mobile robot or a moving vehicle). The work presented in chapter 6 spans both realms by combining information from fixed sensors with information from mobile sensors. To put the proposed framework into context, it is necessary to give an overview and mention related works in: (i) fixed sensor(s) based person detection and tracking, (ii) mobile sensor(s) based person detection and tracking, (iii) sensor fusion modes, and (iv) cooperative systems that try to combine fixed and mobile sensors.

5.1.1 Environment Fixed Sensors

Apparently, research works that use sensors fixed in the environment are vast in number [Hu 2004, Wang 2013]; they include works that use a single classical camera, network of overlapping [Wang 2013] and/or non-overlapping cameras [Meden 2012, Arsic 2008], and a network of heterogeneous sensors (*e.g.*,Laser Range Finders (LRFs) and vision [Cui 2005]). Since the sensors are stationary, simple and fast algorithms like background subtraction and optical flow could be used to detect moving persons within the FOV. Depending on actual sensor configuration, they can encompass wide areas–therefore, provide global perception. They can view and track subjects over a broad area for an extended period of time. But, their main pitfalls include evident dead-spots that could arise from configuration (placement and number of sensors used), possible occlusions, and their passiveness.

5.1.2 Mobile Sensors and Sensor Fusion Modes

On the other hand, mobile robot based systems, as a consequence of their mobility, are generally more suited for surveilling and/or monitoring large areas as they provide a means to reduce the environment structuring and the number of devices needed to cover a given area [Di Paola 2010]. But, multi-person detection and tracking from mobile robots is more challenging due to on-board sensors' motion (during robot mobility), limited FOV of on-board sensors, and limited on-board computational resources. On the other hand, sensors mounted on robots provide localized perception and can pick up details. As a result, robotic based surveillance applications are mostly limited to activities that require close monitoring. They are also suitable for patrolling wide areas owing to their ability to re-position themselves. In addition, they also provide a means for action which can be of paramount advantage for following a target [Germa 2010], intruder intervention [Cory 1999], provision of assistance [Kanda 2010], and possibly physical restraint of an assailant [T-34].

When working with mobile robots, most researchers make use of 2D Laser Range Finders (LRFs) and vision sensors mounted extensively for human detection and tracking. 2D LRFs provide a 2D depth scan of an environment. They have high accuracy, high scanning rates, and are insensitive to lighting conditions. Since they are mostly mounted at a height corresponding to a human leg, person detection proceeds by detecting leg scan patterns in each frame [Arras 2012, Lee 2006]. Some researchers have also mounted LRFs in two layers, scanning at the height of a leg and chest to improve the detection rate, *e.g.*, [Carballo 2009]. Unfortunately, due to their planar scan nature, they are very susceptible to occlusions and are easily fooled by geometrically leg like structures in the environment. They are also not suitable for multi-person tracking with unique identities as they furnish no relevant information for discriminating amongst persons leading to frequent failures in crowded environments. It can be emphasized here that in these scenarios, they ought to be combined with other sensors with more rich perception capabilities. On the contrary, visual sensors provide rich information that capture persons' appearance

well. To detect persons, either background subtraction techniques [Zajdel 2005] or motion segmentation [Chakravarty 2006] can be used from a stationary mobile robot. In case of an active moving robot, recent single frame based approaches like Histogram of Orientation Gradients (HOGs) based person detection [Choi 2013, Mekonnen 2011], face detection [Choi 2013], and though with questionable performance, skin color segmentation [Martin 2006] can be used. For platforms equipped with stereo-vision camera, 3D human like blob segmentation is also a viable option [Beymer 2002]. In effect, vision based multi-person tracking implementations have shown far better results than those based on LRFs owing to rich appearance information and lessened confusion with environment structures. But, they still suffer from narrow FOVs (unless special omni-directional cameras are used), occlusions, and high processing time requirements.

Evidently, most robotic systems are equipped with various sensors and it is only natural to consider fusing the different sensor data to improve individual sensor percepts. The extent of the improvement depends on how well the different sensors complement each other. In the robotic community, fusion of LRF and vision for people detection and tracking has shown to outperform individual counterpart modalities [Kobilarov 2006, Zivkovic 2007]. The fusion, for example, can be done in a sequential manner at the detection level, using the laser hypothesis to constrain the search in the visual data as in [Mekonnen 2011], or in the tracking step [Fritsch 2003]. Variants of Kalman Filters [Bellotto 2009] and Particle Filters [Chakravarty 2006] have been principally used for fusing laser and vision at the tracking step for multi-person tracking. The key interest in laser and vision fusion is combined provision of precise 3D position and rich appearance information which leads to a detection/tracking system with high precision and accuracy. The availability of wide FOV vision system further improves this performance as demonstrated through fusion of a laser with omni-directional cameras [Kobilarov 2006, Chakravarty 2006, Zivkovic 2008].

Furthermore, some researchers have considered fusing vision and audio data [Nakadai 2001, Wu 2009, Fritsch 2004]. Audio data can be used to localize the sound source (possibly a person) and identify the speaker. These are additional features that would enrich the vision data leading to better tracking and identification in crowds. Some works have also considered special sensors like thermal cameras [Treptow 2006] mounted on a mobile robot. Since humans have distinct thermal profile compared to indoor environments, they stand out bright in thermal images which leads to easy detection. But, multi-person tracking in a crowded environment using a thermal camera solely is challenging as human thermal signature is the same for every individual, leading to difficulty in tracked target discrimination amongst each other. [Correa 2012] augmented a thermal camera with classical gray scale camera to realize a system that can detect individuals easily and then use the gray scale image for identification (disambiguation). Another special sensor recently burgeoning is the Kinect [Microsoft 2010]. The Kinect provides an RGB color image and 3D information. In some works, it has been mounted on a mobile robot and used for multi-person perception by fusing the heterogeneous data it provides [Choi 2013, Luber 2011]. Though highly promising, its narrow FOV still remains a problem.

Sensor fusion is certainly not limited to two sensors; depending on availability of sensors and computational time constraint, more sensor data could be fused. For example, Martin *et al.*[Martin 2006] fused LRF, omni-directional camera, and a ring of sonar beams, in a probabilistic aggregation scheme to detect and track individuals in the vicinity of the robot. Zivkovic *et al.*[Zivkovic 2008] combined sensor data from an omni-directional camera, a classical camera mounted on Pan-Tilt-Unit (PTU), and LRF to detect multiple persons using a parts based model. Both cases attest that the plethora of sensors used improve performance well. The improvement comes about mainly because of the complementary nature of the utilized sensors. The rich vision information from cameras can be complemented by employing cameras with different FOVs [Zivkovic 2008],*e.g.*,wide FOV from wall mounted cameras and narrow localized FOV from a camera on a robot.

5.1.3 Environment Fixed and Mobile Sensors

In recent years, researchers have considered surveillance systems that incorporate mobile robots and environment fixed sensors cooperatively. These cooperative surveillance systems combine the merits of fixed and mobile perception modes. They acquire global and wide area perception from the fixed sensors, localized perception and a means for action from the mobile robot. This kind of cooperative systems have the potential to lead to more generic surveillance systems as they can handle various scenarios. Li *et al.*[Li 2008] presented a time-related abnormal events detecting and monitoring system using wireless sensor network and a mobile robot. In their work, intruders are detected using the sensor networks. Upon detection, the mobile robot travels to the position to further investigate the situation locally with its camera. Similarly, in [Chia 2009] three networked wall mounted fixed view cameras and a mobile robot are used to track and follow a target. The target is first detected using the fixed cameras. Once detected, the information is passed onto the robot which navigates to that position and continues to follow the target person. Chakravarty *et al.*[Chakravarty 2009] presented an intruder interception system using external cameras and a mobile robot cooperatively. The external cameras are used to detect an intruder and aid the mobile robot in navigation. The mobile robot, once it has received the location of the intruder, proceeds and intercepts it acting as a means of action to the system. All the above cooperative perception systems portray similar approaches in which perception of interesting targets is initially carried out based on the fixed sensors. The mobile robot's target perception capability is delayed until target presence is communicated to the robot. The perceptual decision making is somewhat decentralized with no data fusion. There is no centralized scheme to collect evidence from the fixed and on-boarded (mobile) sensors to track the targets, rather, either the deported vision, in [Chakravarty 2009], or the mobile robot, in [Chia 2009], does the tracking after the initial target detection. But, an important observation that needs to be made from the related works is data fusion actually leads to robust perception modes.

5.2 Multi-Person Tracking

5.2.1 Overview

Multi-Person Tracking is a special case of Multi-Object Tracking where the tracked targets are persons. Multi-Object Tracking, in its general form, can be interpreted as the process of accurately estimating the state of objects—location, identity, and dynamic configuration—over time from a set of observations. If the tracking uses only past and future observations to determine current state of the objects, it is called causal. On the contrary, if it uses past, present, and future observations, it is then called non-causal. Causal methods are inexpensive and well suited for interactive online usage and lack the ability to correct past errors. Non-causal methods, on the other hand, have the ability to correct past errors, but they are computationally expensive as they usually entail batch processing and are suitable for offline processing like, for example, recorded video annotation/indexing. This brief overview highlights the different aspects and trends of multi-person tracking and is by no means exhaustive. Tracking on ground plane makes it less sensitive to occlusion and allows to consider real information about human dynamics.

In the literature, the majority of multi-person tracking works are based on visual cameras in a video surveillance and robotic contexts [Gabriel 2003]. In general the objectives of multi-person tracking in public place surveillance context are: (1) to correctly estimate the current status of people in the scene, *i.e.*, determine their location, dynamics, and identity; and, (2) to determine a record of trajectories corresponding to each unique observed people over time either on the image plane or ground plane. It should be noted that this is different from human motion

capture which concerns tracking articulated poses of a human. Meeting the above specified objectives is very challenging. Just like the detection problem, it become challenging because of physical variation of targets, deformations due to possible articulations, sensor view point change, sensor motion, background clutter, occlusions, and illumination variations when using vision sensors. Finding a tractable formulation that can cope with these challenges on top of the tracking tasks itself is very difficult, if even possible at all. Whenever possible multi-person tracking system designs should try to anticipate these challenges and try to minimize their impact through careful workarounds. For example, they should put a mechanism to detect when a target is partially/fully occluded so that the tracked target with the same identity could be recovered right when it reappears, *e.g.*, [Gerónimo 2012]. In surveillance context, the principal application of tracking is to provide essential inputs for human activity/event recognition systems.

Multi-person tracking formulation will have to tackle, primarily, the following main issues: (1) how to represent the state of the tracked objects; (2) how to initiate and terminate target tracks automatically; (3) how to model the dynamics of the targets; (4) how to discriminate the targets from each other; and (5) how to associate observations to specific targets, *i.e.*, the *data association problem*.

In the literature, two main paradigms exist for multi-person tracking state representation. The first is a joint representation in which all the states of the tracked targets are joined, as subspaces, to yield a single representation that captures the entire configuration of the tracked persons [Khan 2005, Smith 2005, Isard 2001] and the second is an independent representation whereby each target is represented and consequently tracked independently, *e.g.*, [Breitenstein 2011]. This is effectively assigning an independent single object tracker for each target. The advantage of the joint representation is, should the targets interact, an interaction model can be incorporated in the tracking problem and tackled systematically. On the other hand, for the independent representation, interaction models can not be incorporated directly. It naturally lends itself to ad-hoc solutions based on a higher level supervisor which manages the trackers' behaviors during close-by interaction [Gerónimo 2012, Breitenstein 2011].

Automatic track initialization and termination are necessary functionalities of a fully automated multi-person tracker. Recently burgeoning approaches based on the popular "tracking-by-detection" [Breitenstein 2011, Huang 2008a, Leibe 2007] paradigm tackle this issue reasonably. The basic idea is to employ an automated person detector (addressed in Part I of this manuscript) on each,*e.g.*, [Breitenstein 2011], or sparse,*e.g.*, [Mitzel 2010], tracking frames to initialize a track on a newly detected person, update an already existing track, possible reinitialize a failed tracker, or terminate a track whenever no associated detection occurs for a fixed number of runs. If an automatic detector is not there to do this, the tracking will have to depend on a manually initialized target, which could render track reinitialization after a failure impossible.

Target dynamic model is another issue that needs to be handled. The target dynamic model dictates how the targets evolve in the current time frame from the previous state. In multi-person tracking it is common to consider random walk, *e.g.*, [Smith 2005, Perez 2004], linear autoregressive models with constant velocity, *e.g.*, [Breitenstein 2011], and non-linear models, for example, in the form of social forces [Luber 2010]. Loosely speaking, constant velocity models are suitable when monitoring corridors, parking places, sidewalks, and the like, where people are likely to be heading from one direction to another. Random walks are suitable in situations where people loitering around, like public transportation waiting areas. Non linear models, on the other hand, show promise in very crowded environments where people are likely to show complex social behaviors [Luber 2010]. Some authors, *e.g.*, [Madrigal 2013], have actually considered tracking that incorporates different dynamic models and switches when situation presents itself.

Given the problem is tracking multiple persons, the final tracker should be able to disambiguate each individual. Fortunately, vision based approaches have the rich visual information

at their disposal to define a target appearance/observation model that would help achieve the required disambiguation. In the literature, color histograms in the RGB [Breitenstein 2009], HS+V [Pérez 2002], Lab color spaces [Mitzel 2010], or a combination of these [Zhang 2012] have predominantly been used. It is common to consider a histogram of either the entire human body *e.g.*, [Breitenstein 2009, Gerónimo 2012], or a concatenation of parts-based histograms, *e.g.*, [Smith 2005, Pérez 2002]. The parts-based histograms provide improved intra-class discrimination [Smith 2005, Pérez 2002]. Multi-person tracking based on sensors with no information about the visual appearance of the targets, *e.g.*, LRF, are bound to suffer in this criteria as they can not distinguish observations from nearby targets leading to tracker mix-up [Cui 2008].

The data association problem in multi-person tracking, apparent in the "tracking-by-detection" paradigm, concerns associating existing tracked targets with unique detections, whenever corresponding detections out of the many detections that may arise exist, at each processed time frame. In the literature, the Hungarian algorithm [Kuhn 1955], Joint Probabilistic Data Association Filter (JPDAF) [Rasmussen 2001], and Multiple Hypothesis Tracking (MPT) [Reid 1979] are popular choices. Some authors [Breitenstein 2009, Wu 2007] have found greedy assignment algorithms to work equally well, in fact, at reduced computational burden.

Given the preliminary issues that need to be specified and taken into consideration at design time for any multi-person tracking solution, the next point is what to use to find the actual tracks from a give set of observations. The lions share of the literature in this domain is taken by probabilistic approaches in a Bayesian estimation/inference framework. On fewer accounts deterministic optimization based approaches have been investigated. Examples include, mean shift based multi-person tracker [Beyan 2012], which use mean shift procedure to track each person independently, and level sets [Paragios 2000], which performs detection and tracking of moving objects by the propagation of curves independent for each target. Deterministic approaches, in general, are efficient and quick to converge to a solution. But, they usually run the risk of getting stuck at local minima and for multi-person tracking independent trackers will have to be launched to track each target, *e.g.*, [Beyan 2012]. On the contrary, probabilistic frameworks are quite flexible and principled. They are inherently formulated to take uncertainties and noises in different components of the tracker into consideration. In addition, they are well suited and the popular choice for fusing data from multiple homogeneous or heterogeneous sensors [Smith 2006]. Consequently, we will focus our discussion henceforth on probabilistic approaches specifically based on the Bayesian framework.

As mentioned at the beginning of this section, multi-person tracking is concerned with the problem of tracking a variable number of persons–possibly interacting. The aim is to correctly track and obtain trajectories of the people within the field of view of the utilized sensors. The popular probabilistic approaches in the literature for this include the Multiple Hypothesis Tracker (MHT) [Reid 1979], Joint Probabilistic Data Association Filter (JPDAF)[Rasmussen 2001], joint state [Isard 2001] and independent Particle Filters (PFs) [Breitenstein 2009], and Markov Chain Monte Carlo Particle Filtering (MCMC-PF) [Khan 2005, Smith 2005]. MHT is computationally expensive as the number of hypothesis grows exponentially over time, while JPDAF is applicable to tracking a fixed number of targets. The particle filtering scheme, based on multiple independent PFs per target, suffers from the "hijacking" problem since whenever targets pass close to one another, the target with the best likelihood score takes the filters of nearby targets. The joint state PF scheme–a particle filter with a joint state space of all targets–is not viable for more than three or four targets due to the associated computational requirement. A more appealing alternative in terms of performance and computational requirement is the MCMC-PF. MCMC-PF replaces the traditional importance sampling step in joint PFs by an MCMC sampling step overcoming the exponential complexity and leading to a more tractable solution. For varying number of targets, Reversible Jump Markov Chain Monte Carlo - Particle Filters

(RJMCMC-PFs), an extension of MCMC to variable dimensional state space, has been pioneered to perform successful tracking [Khan 2005, Smith 2005]. When it comes to tracking multiple interacting targets of dynamically varying number [Smith 2007, Khan 2005] have clearly shown that RJMCMC-PFs are more appealing taking performance and computational requirements into consideration. This is also attested in various recent research works, using monocular visual camera [Bardet 2009], or RGB+D cameras [Choi 2013]. Hence, in the next subsequent sections we will focus on presenting the generic formulation of these filters starting from the Bayesian formulation of multi-person tracking.

5.2.2 Bayesian Formulation

To better express multi-person tracking in a Bayesian framework, lets consider tracking as a state estimation problem of a dynamical system that evolves over time. If X_t denotes the state of the system at time t, then it is possible to completely define the system by its prior distribution $p(X_0|Z_0) \equiv p_0(X_0)$, dynamical model, equation 5.1, and measurement model, equation 5.2.

$$X_t = f_t(X_{t-1}, \mathbf{v}_{t-1}) \tag{5.1}$$
$$Z_t = h_t(X_t, \mathbf{n}_t) \tag{5.2}$$

Where $\mathbf{v_t}$ and $\mathbf{n_t}$ represent the process and measurement noise respectively.

Given this system, the goal is to estimate a distribution of the system state histories up to time t, $p(X_{0:t}|Z_{1:t})$, given the measurements, $Z_{1:t} = \{Z_1, Z_2, ..., Z_t\}$. This posterior on the state chain up to time t can be expressed as in equation 5.3 considering a first-order Markovian assumption, which dictates the current system state at time t only depends on the previous state at $t-1$, and assuming the measurements are conditionally independent given the state.

$$\begin{aligned} p\left(X_{0:t}|Z_{1:t}\right) &= \frac{p(Z_t|X_{0:t}, Z_{1:t})p(X_{0:t}|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\ &= \frac{p(Z_t|X_t)p(X_{0:t}|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\ &= \frac{p(Z_t|X_t)p(X_t|X_{t-1})}{p(Z_t|Z_{1:t-1})} p(X_{0:t-1}|Z_{1:t-1}) \end{aligned} \tag{5.3}$$

At each time frame t, we are mostly interested in estimating the current state distribution given the measurements, $p(X_t|Z_{1:t})$, which is shown equation 5.4.

$$\begin{aligned} p(X_t|Z_{1:t}) &= \frac{p(Z_t|X_t)p(X_t|Z_{1:t-1})}{p(Z_t|Z_{1:t-1})} \\ &= \frac{p(Z_t|X_t)\int p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1}}{p(Z_t|Z_{1:t-1})} \\ &= \frac{p(Z_t|X_t)\int p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1}}{\int p(Z_t|Z_{1:t-1}, X_t)p(X_t|Z_{1:t-1})dX_t} \\ &= C\ p(Z_t|X_t)\int_{X_{t-1}} p(X_t|X_{t-1})p(X_{t-1}|Z_{1:t-1})dX_{t-1} \end{aligned} \tag{5.4}$$

Where C a normalization constant that insures $p(X_t|Z_{1:t})$ is a probability distribution.

Equation 5.4 shows state posterior distribution expressed in a recursive Bayesian filtering formulation. At each time step, the filter computes the posterior distribution, as a new measurement comes, in two steps: prediction and update. In the prediction step, it uses the system dynamic model (also called motion model) combined with the previous time posterior distribution to predict the new state. In the update step, it uses the last likelihood measure, $p(Z_t|X_t)$, to update the state belief from the prediction step.

In our case, we are interested in multi-person tracking using a joint space representation. In this case the state of the system at time t, X_t, represents the entire configuration of the targets concatenated in one variable. Hence, $X_t = \{I_t, x_{(t,i)}\}_{i\in\{1,2,\ldots,I_t\}}$ where I_t denotes the number of tracked persons and $x_{(t,i)}$ denotes the state vector of individual persons indexed by i, at time t. The integrals in the recursive Bayesian filter equation, equation 5.4, are analytically intractable. In the literature of multi-person tracking, the popular choice is to use particle filters based on sampled approximations of the posterior. With the joint state configuration, a straight forward use of the joint particle filter based on Important Sampling suffers from exponential complexity of the number of tracked targets making it unusable for more than three or four targets [Smith 2007, Khan 2005, Khan 2003]. On the other hand, Markov Chain Monte Carlo based particle filters that rely on the Metropolis-Hastings (MH) algorithm [Hastings 1970] for sample generation are suited for tracking a number of targets in a computationally tractable manner [Smith 2007, Khan 2005].

5.2.3 MCMC- and RJMCMC-Particle Filters

In this section two Particle Filter variants based on MCMC and RJMCMC sampling are presented. The MCMC-PF is suitable for tracking fixed number of persons, whereas RJMCMC-PF can handle varying number of persons. Both filters employ a joint state representation for multi-person tracking.

5.2.3.1 MCMC-PF

The MCMC-PF starts out by approximating the posterior at time $t-1$ using a set of N discrete unweighted samples: $p(X_{t-1}|Z_{1:t-1}) \approx \{X_{t-1}^{(n)}\}_{n=1}^{N}$. With this, the posterior at the current time frame is approximated with equation 5.5.

$$p\left(X_t|Z_{1:t}\right) \quad \approx C\ p(Z_t|X_t) \sum_{n=1}^{N} p(X_t|X_{t-1}^{(n)}) \tag{5.5}$$

MCMC-PF then defines a Markov Chain over the state configuration so that the stationary distribution of the chain approximates the posterior distribution (equation 5.5). Algorithm 5.1 details the entire MCMC-PF algorithm for tracking a fixed number of M targets. In it, steps 4 to 12 pertain to the Metropolis Hastings (MH) algorithm for sampling from the chain. The MH step requires definition of the proposal distribution, $Q(\cdot)$, from which the particles are actually sampled and the acceptance ratio, β, term. An important simplification made by [Khan 2005] is to only update the state of a single target in each iteration of the MH. This is a key point that makes the filter efficient. In addition, updating only a single target means the likelihood measures and the proposal distribution only vary in those dimensions which leads to canceling of terms simplifying the evaluation of the acceptance ratio.

The algorithm starts out by initially selecting a random particle from the particle set in the previous time frame and applying the motion model to all targets, step 2-3. Then it starts the

MH cycle constructing a Markov Chain and sampling from it. It proposes a new particle sampled from the proposal distribution $X^* \sim Q\left(X^*|X_t^{n-1}, Z_t\right)$, steps 6-8. This is achieved by randomly selecting a single target x_j from the particle at the previous MH iteration, $X_t^{(n-1)}$. Since only a single target is considered, the proposal distribution simplifies to being a function of the target and measurement input (if measurement is taken into consideration at the sampling stage), step 8. Then, the acceptance ratio is computed for the proposed update, step 10. The proposed particle is then accepted with probability β or rejected (in which case the particle configuration in the previous MH iteration is repeated). Bear in mind here that the MH iteration is performed for $N_T N + N_B$ times. The N_B represents the *burn-in* iterations of which the samples are discarded to avoid any bias from the sampling starting point. N_T is the number of *thin-out* iterations used to reduce correlation between samples; the samples in between are again discarded. Finally, a point estimate can be derived by computing the expectation of X_t according to the posterior distribution, step 15. The proposal distribution function is a design choice. It can be designed to incorporate measurement information, or could be as simple as perturbing a target state according to a zero-mean normal distribution.

Algorithm 5.1 MCMC-PF based Multi-person Tracking (fixed M number of targets)

1: **procedure** MCMC_TRACK($\{X_{t-1}^{(n)}\}_{n=1}^{N}$; Z_t)
2: · **Init:** pick a random particle from the set $\{X_{t-1}\}$ and apply motion model.
3: $X_t^0 \sim p\left(X_t|X_{t-1}^{(r)}\right)$
4: **for** $n \leftarrow 1$ to $N_T N + N_B$ **do**
5: · Propose a new particle $X^* \sim Q\left(X^*|X_t^{(n-1)}, Z_t\right)$
6: - Randomly select a target x_j from $X_t^{(n-1)}$ where $j \in \{1, .., M\}$
7: - Select x_j^*, a random subspace corresponding to x_j in the particle set at $t-1$
8: - Resample $x_j \sim Q\left(x_j|x_j^*, Z_t\right)$
9: · Compute acceptance ratio:
10: $\beta = \min\left(1, \frac{p(Z_t|X^*)\cdot Q\left(X_t^{(n-1)}|X^*, Z_t\right)}{p\left(Z_t|X_t^{(n-1)}\right)\cdot Q\left(X^*|X_t^{(n-1)}, Z_t\right)}\right)$
11: · Accept X^* with probability β, *i.e.*, $X_t^{(n)} \leftarrow X^*$, otherwise reject and
12: set $X_t^{(n)} \leftarrow X_t^{(n-1)}$
13: **end for**
14: · Discard the first N_B samples of the chain and retain only every N_T^{th} particle
15: · Point estimate, $\hat{X}_t := E_{p(X_t|Z_{1:t})}[X_t]$
16: **return** $\{X_t^{(n)}\}_{n=1}^{N}$ and $\hat{X}_t$
17: **end procedure**

Interaction Term: In this MCMC-PF formulation, it is straight forward to incorporate an interaction model between the tracked targets thanks to the joint state configuration representation. If we denote an interaction term $\Psi(X_t)$ to model the interaction between the states of the different targets, the dynamic model can be readily altered to accommodate this as $\Psi(X_t)p(X_t|X_{t-1})$. This leads to a posterior approximation of $p(X_t|Z_{1:t}) \approx C\, p(Z_t|X_t)\Psi(X_t)\sum_n p\left(X_t|X_t^{(n-1)}\right)$. Accordingly, the acceptance ratio computation will be done using equation 5.6.

$$\beta = \min\left(1, \frac{p\left(Z_t|X^*\right)\Psi(X^*)Q\left(X_t^{n-1}|X^*, Z_t\right)}{p\left(Z_t|X_t^{(n-1)}\right)\Psi(X_t^{(n-1)})Q\left(X^*|X_t^{n-1}, Z_t\right)}\right) \tag{5.6}$$

The presented MCMC-PF is an interesting solution when tracking a fixed number of possibly interacting targets. The possibility for explicit interaction model inclusion with the filter's efficient sampling makes has made it more appealing to some authors. For example, [UrRehman 2012] used MCMC-PF so as to include an interaction model to detect automatic occlusion and reinitialize the tracked region in multiple head tracking. But, the ability to track a fixed number of targets has decreased the attention it ought to get paving the way to its RJMCMC-PF generalization. In the robotic context, Tanaka and Kondo [Tanaka 2004] have used this filter based on vision to track a fixed number of targets in office environments.

5.2.3.2 RJMCMC-PF

The MCMC-PF presented previously is suitable for tracking only a fixed number of targets. But, in reality, we are interested in a multi-person tracker that can track varying number of people as people come in and go out of a surveilled area. Khan *et al.* [Khan 2005] and Smith *et al.* [Smith 2005], independently proposed the Reversible Jump Markov Chain Monte Carlo Particle Filter (RJMCMC-PF), sometimes also called trans-dimensional MCMC-PF. This is a generalization of the MCMC-PF. Its main difference is that, it has a variable dimension state representation. It uses a set of moves, m, to either change the dimension of the state, by increasing or decreasing it, or leave it unchanged, according to a prior move distribution, q_m on the move types. Each move is associated with a move specific proposal distribution, $Q_m()$. Algorithm 5.2 details the RJMCMC-PF based multi-target tracker. An important point is the reversibility of each move type. Each move m must have a reverse move m^* that assures reversibility so that detailed balance will be achieved and the chain will converge to the desired stationary distribution [Khan 2005] (for a move that does not change the dimension, the reverse move type is itself). The algorithm starts out, like the MCMC-PF, with a randomly sampled particle and applies the motion model to each target, step 3. It then starts the RJMCMC sampling cycle. It first samples a move type according to the prior move distribution. It then samples a new particle, X^*, from the move specific proposal distribution, step 6-7. Again on each Markov Chain iteration, changes are done to a single target to make the sampling efficient. Then, it computes the acceptance ratio, β, taking considering the move specific proposal distribution, $Q_m()$, and its reverse move proposal distribution, $Q_{m^*}()$, step 10. The proposed particle is accepted with probability β or otherwise rejected. Particles used both for the *burn-in* and *thin-out* are discarded leaving N unweighted samples to represent the posterior.

Similar to the MCMC-PF, an interaction term of the form , $\psi(X_t)$ can be incorporated. In this case, the acceptance ratio computation is performed according to equation 5.7. In multiple counts, a dynamically constructed pairwise Markov Random Field (MRF) that models the interactions between nearby targets has been privileged by many researchers [Khan 2005, Smith 2005, Choi 2013].

$$\beta = \min\left(1, \frac{p\left(Z_t|X^*\right)Q_{m*}\left(X_t^{(n-1)}|X^*, Z_t\right)q_{m*}\Psi\left(X^*\right)}{p\left(Z_t|X_t^{(n-1)}\right)Q_m\left(X^*|X_t^{(n-1)}, Z_t\right)q_m\Psi\left(X_t^{n-1}\right)}\right) \tag{5.7}$$

Algorithm 5.2 RJMCMC-PF based Multi-person Tracking (variable number of targets)

procedure RJMCMC_TRACK($\{X_{t-1}^{(n)}\}_{n=1}^{N}$; Z_t)
  Init: pick a random particle from the set $\{X_{t-1}\}$ and apply motion model.
    $X_t^0 \sim p\left(X_t | X_{t-1}^{(r)}\right)$
  for $n \leftarrow 1$ to $N_T N + N_B$ **do**
    · Sample a move $m \sim q_m$
    · Propose a new particle $X^* \sim Q_m\left(X^* | X_t^{(n-1)}, Z_t\right)$
      - Modify $X_t^{(n-1)}$ in accordance with the selected move, m,
      and move specific proposal distribution, $Q_m(\cdot)$
    · Compute acceptance ratio:
      $$\beta = \min\left(1, \frac{p(Z_t|X^*) q_{m*} Q_{m*}\left(X_t^{(n-1)} | X^*, Z_t\right)}{p\left(Z_t | X_t^{(n-1)}\right) q_m Q_m\left(X^* | X_t^{(n-1)}, Z_t\right)}\right)$$
    · Accept X^* with probability β, *i.e.*, $X_t^{(n)} \leftarrow X^*$, otherwise reject and
      set $X_t^{(n)} \leftarrow X_t^{(n-1)}$
  end for
  · Discard the first N_B samples of the chain and retain only every N_T^{th} particle
  · Identify the mode of the particle configurations and define $\tilde{X}_t$ as containing
    only those particles conforming with the mode configuration
  · Point estimate, $\hat{X}_t := E_{p(X_t|Z_{1:t})}[\tilde{X}_t]$
  return $\{X_t^{(n)}\}_{n=1}^{N}$ and $\hat{X}_t$
end procedure

In general, owing to its ability to accommodate a variable number of targets, interaction model, and its efficient sampling strategy, RJMCMC-PF has been used for multi-person tracking, either in the image plane, or 3D world, from static monocular cameras [Smith 2005], robotic context using RGB+D sensor [Choi 2013], and environment fixed camera networks [Yao 2009]. Bardet *et al.* [Bardet 2009], have also used to jointly track and classify objects and light sources validated via multiple vehicle and pedestrian tracking under different variable illumination. Consequently, the RJMCMC-PF is adopted as the de-facto framework for multi-person tracking in our subsequent works (presented in chapter 6).

5.2.4 Evaluation Metrics

Evaluation of multi-person tracking systems focuses on four main aspects: (1) Whether it can correctly detect the presence and/or absence of targets in the scene (Miss Rate); (2) How well it can filter out false alarms (False Positives); (2) How precise the tracking estimates are (with respect to the true targets' positions); (3) How well it can keep track of a unique target maintaining its identity consistently and, especially, without any mix-up with other targets. In the literature, the most frequently used set of metrics that take these points into consideration are the CLEAR MOT metrics [Bernardin 2008], which are the de-facto for evaluating multi-object tracking. The two most important succinct quantities are the **MOTP** and **MOTA**. Below each of these metrics along with rudimentary metrics used to compute them are explained.

- Tracking Success Rate (TSR): given by $\frac{1}{J_T}\sum_{t,j}\delta_{t,j}$ where $\delta_{t,j} = 1$ if target j is tracked at time frame t, else 0. $J_T = \sum_{t,j} j_t$, and j_t represents the number of persons in the tracking area at time frame t.

- Miss Rate (MR): is the ratio of misses in the sequence, computed over the total number of objects in all frames, i.e. $\frac{1}{J_T}\sum_{t,j}\delta_{t,j}$ with $\delta_{t,j}=1$ if the target j in the area is not tracked by any tracker at time frame t, else 0.
- Ghost Rate (GR): computes the number of candidate targets over no target (ghosts) averaged over the total number of targets in the dataset, i.e. $\frac{1}{J_T}\sum_{t,j}\delta_{t,j}$ with $\delta_{t,j}=1$ if tracked target j is a ghost at time frame t, else 0.
- Mismatch: mismatch error occurs when an existing tracked target is initialized as a new target or takes the id of another existing tracked target. Mismatch is computed by counting the number of mismatch errors that occur through out the dataset. Mismatch_t specifies the mismatch at time frame t.
- Multiple Object Tracking Precision (MOTP): measures how precisely the targets are tracked as the sum of the error between tracker position estimate and ground truth averaged over the total number of correct tracks made. If the tracking is done on the image plane, it is expressed in pixel units, and if it is done in real world coordinates (3D or ground plane), it is expressed in metric units (usually in centimeters).
- Multiple Object Tracking Accuracy (MOTA): is an accuracy metrics computed by taking the total errors (Miss Rate, False Positive (FP), and Mismatch) in each time frame t into consideration.
$$MOTA = 1 - \frac{\sum_t(MR_t + FP_t + \text{Mismatch}_t)}{J_T} \tag{5.8}$$
- Id Swap: this criterion quantifies how many times an id switch between two different tracked targets occurred. It is represented as $\sum_t\sum_{i,j}\delta_{i,j}$, where $\delta_{i,j}=1$ when an id switch occurs between tracked target i and j in time frame t, otherwise it is 0,

In the above criteria, an observation that should be made is the delineation of Mismatch and Id Swap. The Mismatch criterion counts the number of mismatches that occur for all tracks. It counts both initialization of an already tracked target with a new identifier and id swap between tracked targets as a mismatch error.

5.3 Conclusion

In this chapter a concise overview on different people detection and tracking systems have been presented. This has been followed by presentation of the multi-person tracking formulation in a recursive Bayesian formulation and popular filter choices based on Markov Chain Monte Carlo sampling. As has been presented, the RJMCMC-PF is well suited for tracking a variable number of interacting targets in a tractable manner. This tracking framework is adopted and used in the subsequent chapter to realize a multi-person tracking system fusing information from wall mounted cameras and sensors on a mobile robot. The framework is also adopted for implementing a multi-person tracking system based with a mobile robot equipped with a laser range finder and spherical camera whose evaluation is currently on-going.

Chapter 6

Implementation of A Cooperative Perception System

Contents

6.1 Introduction . . . 102

6.2 Framework and Architecture . . . 102

6.2.1 Environment Configuration . . . 102

6.2.2 System Block Diagram . . . 103

6.2.3 Environment Calibration . . . 103

6.3 Perceptual Components . . . 105

6.3.1 Multi-person Detection . . . 105

6.3.2 Multi-person Tracking Implementation . . . 108

6.4 Robot Navigation Aspects . . . 113

6.4.1 Personal Space Model . . . 113

6.4.2 Nearness Diagram (ND) Navigation . . . 114

6.5 Evaluations and Results . . . 114

6.5.1 Off-line Evaluation . . . 114

6.5.2 On-line Evaluation . . . 118

6.6 Towards a Self-Contained Robotic Perceptual System . . . 124

6.7 Discussions . . . 125

6.8 Conclusion . . . 126

6.1 Introduction

This chapter presents a cooperative perception system between environment fixed cameras and a mobile robot for detecting and tracking people in a monitored environment. The cooperation is carried out in a centralized manner, all data from the different sensors are gathered in a central processing unit that does the data fusion and inference using all information at once. As has been mentioned in chapter 1, this kind of perceptual modalities are advantageous in monitoring sensitive areas that require increased accuracy, as it utilizes redundant data, for example near a security checkpoint in the airport terminal briefly discussed in section 1.1. The chapter also gives outline/insight of an ongoing development that uses sensors mounted on a mobile robot cooperatively to realize a self-contained perception system (section 6.6).

The proposed cooperative perception involves a mobile robot deployed in public people occupied environments. The advantages garnered by this introduction of a mobile robot come with one major constraint—safety considerations during the robot's navigation in public environments. We try to mitigate this constraint by utilizing the perceptions from the cooperative perceptual system to define specialized security zones around each person that lead to safe robotic navigation. Briefly speaking, this chapter makes two core contributions. First, it proposes and validates a centralized cooperative framework and data fusion scheme between environment fixed fixed-view cameras and sensors embedded on a mobile robot to track multiple passers-by in a surveilled environment. Second, it proposes a methodology to utilize the perceived people's information to realize safe robot navigation in populated environments and demonstrates it with live robotic experiments. The proposed perceptual functionalities partly rely on the people detector implemented and presented in Part I of this thesis. The improvements brought up by our proposed detectors are also highlighted in the on-line experiments.

This chapter mostly focuses on implementation details and evaluations as most of the background material has been presented in chapter 5. Consequently, it starts with a presentation of the framework and architecture of the proposed system in section 6.2. It then presents implementation details of the perceptual components, detection and tracking functionalities, in section 6.3. Robotic navigation considerations are discussed in section 6.4. The different off-line and on-line evaluations carried out along with obtained results are discussed in section 6.5. In section 6.6 outlines to an ongoing self-contained robotic perception system are briefly presented. Finally, the chapter culminates with discussions in section 6.7 and concluding remarks in section 6.8.

6.2 Framework and Architecture

Our main objective is to correctly detect and track people in a surveilled area using a cooperative perception system made up of two fixed view wall mounted cameras and sensors on-board a mobile robot. This section presents a description of the proposed system and corresponding environment configuration.

6.2.1 Environment Configuration

Our cooperative framework is made up of a mobile robot and two fixed view wall-mounted RGB flea2 cameras, denoted as cameras c_1 and c_2. The cameras have a maximum resolution of 640×480 pixels and are connected to a computer *via* a fire-wire cable. The mobile robot, Rackham, shown in figure 6.1 is an iRobot B21r mobile platform. Rackham has various sensors, of which its SICK Laser Range Finder (LRF), positioned $38cm$ above the ground and with a 180^o FOV, Micropix digital camera mounted on a Directed Perception pan tilt unit (PTU), and an

omni-directional RF system custom-built in the lab for detecting RF tagged person all around the robot [Germa 2010], are utilized in this work. Rackham has two PCs (one mono-CPU and one bi-CPU PIII running at 850 MHz) and a Wireless Ethernet. Figure 6.1 shows the hardware aspect of our framework. Communication between the mobile robot and the computer hosting the cameras is accomplished through a wi-fi connection. Rackham's software architecture is based on the **G**eno**M** architecture for autonomy [Alami 1998]. All its functionalities have been embedded in modules created by **G**eno**M** using C/C++ interface.

Figure 6.1 illustrates a schema of the environment and the mobile robot. Communication between the mobile robot and the computer hosting the cameras is accomplished through a wi-fi connection.

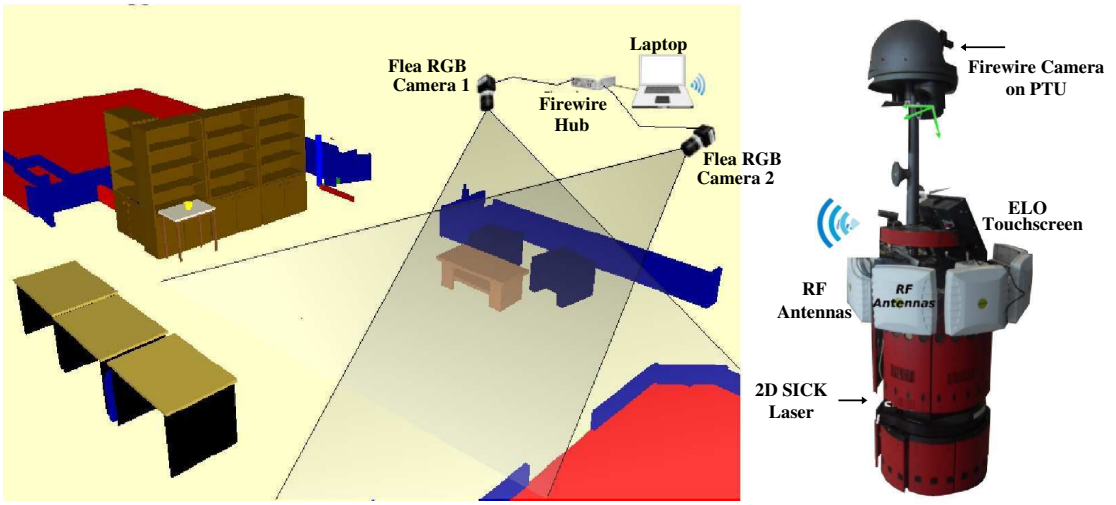


Figure 6.1: Cooperative perceptual platform; wall-mounted cameras (with rough positioning and fields of view) and Rackham, the mobile robot.

6.2.2 System Block Diagram

Figure 6.2 shows block diagram of the envisaged cooperative people detection and tracking framework using wall mounted fixed-view cameras a mobile robot. It has two main parts. The first part deals with automated multi-person detection. The second part is dedicated for multi-person tracking. It takes all detections as input and fuses them in a Particle Filtering framework (discussed in section 6.3.2). Each of these parts are discussed in detail in subsequent sections. It is worth mentioning here that the entire system is calibrated with respect to a global reference frame (described in section 6.2.3). Both the intrinsic and extrinsic parameters of the fixed cameras are known and in addition the mobile robot has a localization module that localizes its pose with respect to the reference frame using laser scan segments [Clodic 2006].

6.2.3 Environment Calibration

This framework involves sensors deployed in different parts of the environment with some even being mobile. As a result the cooperation relies on correct environment calibration. Here, the notations assigned and procedures employed for the calibration are presented.

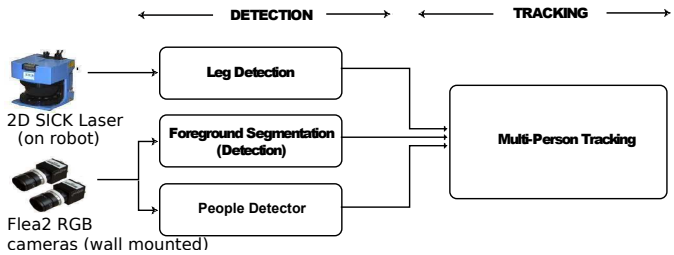


Figure 6.2: Cooperative people detection and tracking framework.

Let the notation ${}^{E}T_{D}$ express the transformation matrix that describes frame D with respect to frame E and hence can be used to determine homogeneous coordinates of all coordinate points, expressed with respect to frame D, with respect to frame E by direct multiplication. For the configuration shown in figure 6.3, the goal of the calibration is to determine such transformation matrix for both cameras, c_1 and c_2, and the mobile robot with respect to the world coordinate frame $\{W\}$, *i.e.*, ${}^{W}T_{C_1}$, ${}^{W}T_{C_2}$, and ${}^{W}T_{R}$.

Lets begin with camera c_1 with an intrinsic matrix of A_1. Now a point P in the world expressed with respect to camera c_1, ${}^{C_1}P$, its projection on the image plane, ${}^{c_1}p$, is determined as, $s{}^{c_1}p = A_1{}^{C_1}P$, where s is a scaling factor. But, if point P is initially expressed with respect the world frame, ${}^{W}P$, it needs to be first expressed with respect to c_1 as ${}^{C_1}P = {}^{C_1}T_{W}\,{}^{W}P$. ${}^{C_1}T_{W}$ is a transformation matrix of the form $[R|t]$ with rotation and translation components that expresses the world frame with respect to c_1. ${}^{C_1}T_{W}$ is called the extrinsic camera transformation matrix.

To determine ${}^{C_1}T_{W}$, we can use a calibration checkerboard of known dimensions. This checkerboard is then placed on the floor and its pose measured accurately with respect to the world frame. This is expressed with ${}^{W}T_{B}$. Similarly, ${}^{C_1}T_{B}$ expresses its pose with respect to camera c_1 frame.

All interior corner points of the checkerboard pattern can be expressed accurately with respect to the board's frame. For a corner point P_i, this is ${}^{B}P_{i} = {}^{B}[X_i \;\; Y_i \;\; Z_i \;\; 1]^{T}$ in homogeneous coordinate. Consequently, let matrix ${}^{B}\mathbf{P} = [{}^{B}P_1\,{}^{B}P_2 ... {}^{B}P_N]$ represent all points of the interior checkerboard corners, N being the total number of such corners. These points can be succinctly expressed with respect to camera c_1 by ${}^{C_1}\mathbf{P} = {}^{C_1}T_{B}\,{}^{B}\mathbf{P}$ and their corresponding projection on the image of camera c_1 using equation 6.1.

$$s{}^{C_1}\mathbf{p} = A_1{}^{C_1}T_{B}\,{}^{B}\mathbf{P} \tag{6.1}$$

But, in equation 6.1, ${}^{C_1}\mathbf{p}$ can be determined from the image coordinates of the checkerboard corners; ${}^{B}\mathbf{P}$ can be easily measured; and, A_1 can be determined by intrinsic camera calibration. The only unknown in this equation is ${}^{C_1}T_{B}$. This transformation matrix, hence, can be estimated iteratively using Levenberg-Marquardt optimization by finding such a pose that minimizes re-projection error, *i.e.*, the sum of squared distances between the observed projections of the checkerboard corner on the provided image and the projected (using the estimated transformation matrix) object points. The actual extrinsic camera transformation ${}^{C_1}T_{W}$ is then determined as ${}^{C_1}T_{B}\,{}^{B}T_{W} = {}^{C_1}T_{B}[{}^{W}T_{B}]^{-1}$. These steps are performed again using the image from camera c_2 to determine ${}^{C_2}T_{W}$. The mobile robot on the other hand, has an active localization module that uses an a priori built map of the environment from laser scans to localize the robot with respect to the world frame using the laser scan data. This module provides ${}^{W}T_{R}$. Whenever there is a detection in the images from the cameras, the detection can be projected on to the floor, by intersecting the vectors emanating from the pixel positions of the legs with the floor

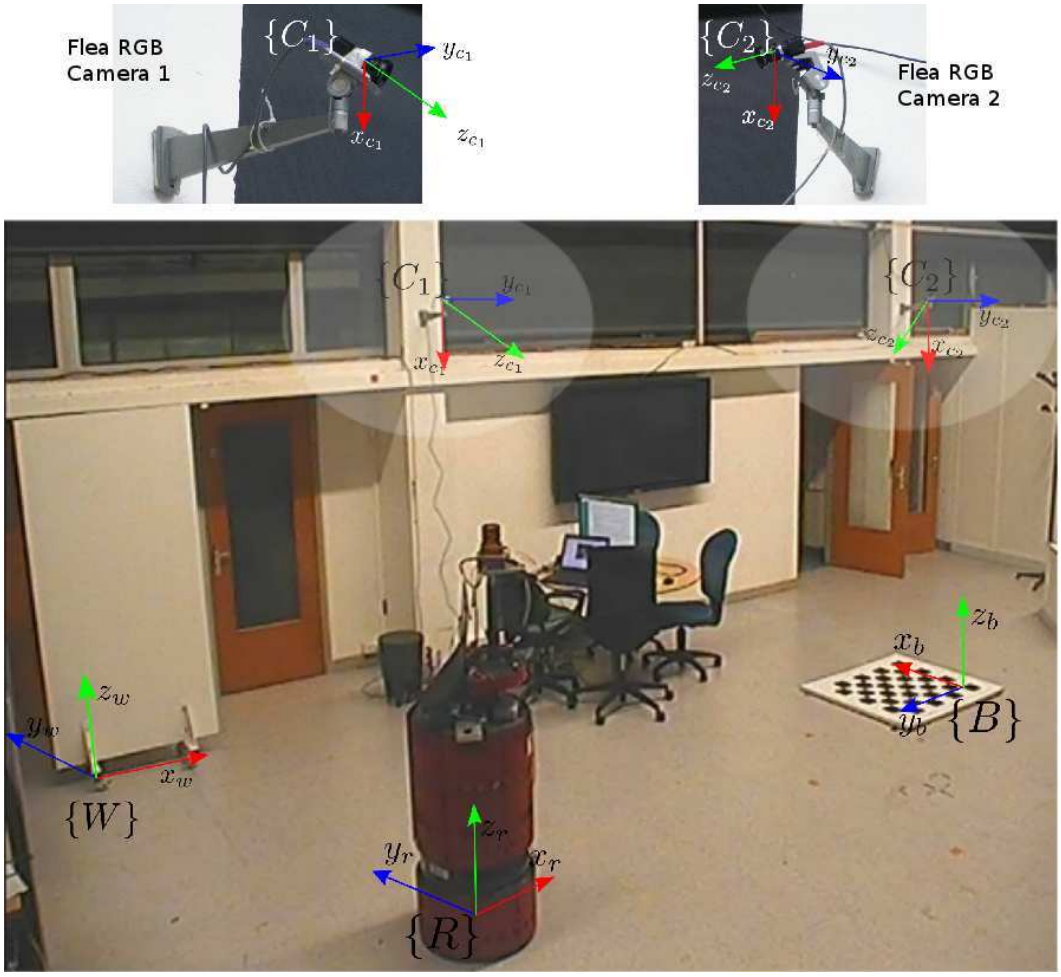


Figure 6.3: Cooperative system environment calibration.

equation; this provides an accurate (x, y) position of the targets with respect to the world frame. The calibration parameters determined in accordance with the presentation in this section are presented in appendix B.

6.3 Perceptual Components

6.3.1 Multi-person Detection

The perceptual functionalities of the entire system are based on various detections. The detection modules are responsible for automatically detecting people in the area. Different people detection modalities are utilized depending on the data provided by each sensor.

6.3.1.1 Leg Detection with LRF

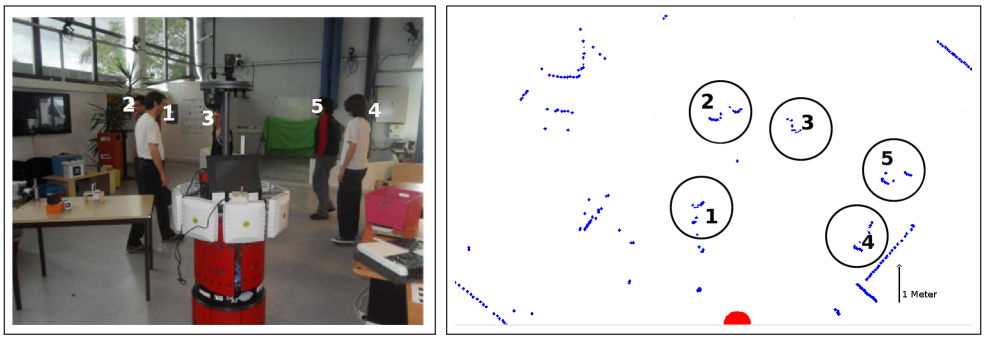
Laser Range Finders (LRFs) have become attractive tools in the robotics area for environment detection due to their accuracy and reliability. As the LRFs rotate and acquire range data, they will have distinct scan signatures corresponding to the shape of an obstacle in the scan region. In our case, the LRF provides horizontal depth scans with a 180^o FOV and 0.5^o resolution at a height of $38cm$ above the ground. Person detection, hence, follows by segmenting leg patterns within the scan. In our implementation a set of geometric properties characteristic to human legs and outlined in [Xavier 2005] are used. Figure 6.4 shows an instance of a scan with leg signatures circled and the actual human-robot situation. The detection proceeds in three steps:

1. Blob segmentation. All sequential candidate scan points that are close to each other are grouped to make blobs of points. The grouping is done based on the distance between consecutive points.

2. Blob filtering. The blobs formed are filtered using geometric properties outlined in [Xavier 2005]. The filtering criteria used are: *Number of scan points*, *Mid point distance*, *Mean Internal Angle and Internal Angle Variance*, and *Sharp structure removal*.

3. Leg formation. All the blobs that are not filtered out by the above stated requirements are considered to be legs. Each formed leg is then paired with a detected leg in its vicinity (if there is one). The center of the paired legs makes the position of the detected human.

Each detected person has an associated appearance representation obtained by projecting a rectangular region, corresponding to an average person, onto the wall mounted camera images thanks to the fully calibrated system. The appearance is captured in the form Hue-Saturation+Value (HS+V) [Pérez 2002] histogram. Individual histograms are obtained from the two cameras, of course if the detection is within the field of view, and are treated separately.



(a) (b)

Figure 6.4: LRF scan illustrations showing the human-robot situation in (a) and the associated laser scan in (b). Scans corresponding to legs are shown circled. Rackham is shown as the red circle in (b).

6.3.1.2 Foreground Segmentation (Detection) with Wall Mounted Cameras

The two wall mounted cameras with partially overlapping FOV provide a video stream of the area. One person detection mode employed is foreground segmentation using background subtraction as these cameras are static. To accomplish this, a simple Σ-Δ background subtraction technique [Manzanera 2007] is used. After a series of morphological operations, only foreground

blobs with an aspect ratio comparable to an average human being are kept and treated as detected persons. The mobile robot is masked out of the foreground images using its position from its localization module. The detections are projected to yield ground positions, $(\mathsf{x}, \mathsf{y})_G$, with associated color appearance information, in the form of HS+V histograms, of individuals in the area and are then passed along to the passers-by tracking module. Figure 6.5b shows sample foreground segmented image with bounding box to show detected humans from both cameras.

(a) (b) (c)

Figure 6.5: Sample images from the two wall mounted cameras. (a) shows the actual feed, (b) shows the segmented foreground/background image based on Σ-Δ background subtraction technique with bounding boxes, and (c) detection from the people detector module, in this specific case HOG detector [Dalal 2005] is used.

6.3.1.3 People Detection with Wall Mounted Cameras

Similar to the foreground segmentation step, a people detector is used to automatically detect persons in the surveilled area using the feed from the wall mounted cameras. In this component, any of the holistic person detectors discussed in section 2.2.4 could be used. We actually use Dalal and Triggs HOG detector [Dalal 2005] and the HOG-BIPBoost variant presented in section 3.8 later with the on-line experiments (section 6.5.2) to show the effect of the detector speed improvement our proposed detector has on the overall robotic functioning. Both detectors make no assumption of any sort about the scene or the state of the camera (mobile or static). In subsequent sections, we will refer to this detector as HOG detector without any loss of generality until there is a need to make an explicit distinction. Again the detections are passed to the multi-person tracking module once projected into ground position, $(\mathsf{x}, \mathsf{y})_G$, with an associated HS+V histogram. Sample HOG based person detections are shown in figure 6.5c; corresponding sample HS+V histograms computed as in [Pérez 2002] are shown in figure 6.6. The histograms have an 8×8 HS bin and 8 V bin. They are shown unrolled in a single dimension to ease visualization.

To summarize, five sets of detections are produced for the multi-person tracking module, namely: one from the LRF (l), two from the wall mounted cameras via foreground segmentation ($fseg_{c_1}, fseg_{c_2}$), and another two via HOG detection from the same cameras

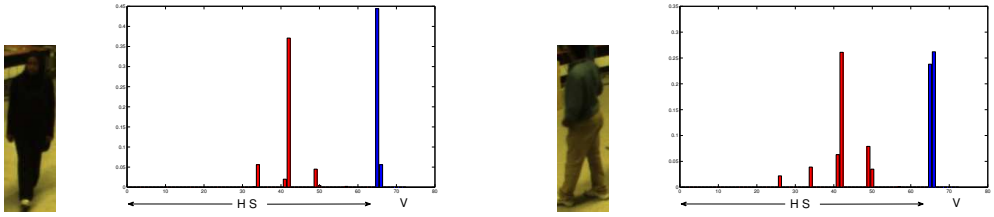


Figure 6.6: HS+V histograms computed for two targets. The histograms have 8x8 HS bin and 8 V bin. They are shown here unrolled in a single dimension.

(hog_{c_1}, hog_{c_2}). Hence, the complete set of detections passed along at time t is denoted as $\{z^d_{t,j} : d \in \{l, fseg_{c_1}, fseg_{c_2}, hog_{c_1}, hog_{c_2}\}, j \in \{1, .., N_d\}\}$ where N_d represents the number of detections in the d^{th} detector and each z denotes a detected person position on the ground floor $(\mathsf{x}, \mathsf{y})_G$.

6.3.2 Multi-person Tracking Implementation

As clearly justified in section 5.2, we will use the RJMCMC-PF framework for the purpose of multi-person tracking. Our RJMCMC-PF tracker is driven by the heterogeneous detectors that provide ground position of individual persons and their corresponding appearance information (section 6.3.1). The actual detectors are: the LRF based person detector, the foreground segmentation (detection) from each wall mounted camera, and the HOG based person detector on each wall mounted camera. The multi-person tracking is performed on the ground plane. The generic RJMCMC-PF tracking algorithm is presented in detail in section 5.2. Here, specific implementation choices crucial to any RJMCMC-PF are presented below. The choices pertain to: the target state space; the jump moves, m, and associated distribution, q_m; proposal move distribution, $Q_m(\cdot)$, associated with each move; the observation likelihood, $p(Z|X)$; and the interaction model, $\Psi(\cdot)$.

Our choice of these components that are crucial to any RJMCMC-PF implementation are discussed below. The complete multi-person tracking algorithm based on RJMCMC-PF is presented in algorithm 6.1.

6.3.2.1 State Space

The state vector of a person i in hypothesis n at time t is a vector encapsulating the id and $(\mathsf{x}, \mathsf{y})_G$ position of an individual on the ground plane with respect to a defined coordinate base, $x^n_{t,i} = \{Id_i, \mathsf{x}^n_{t,i}, \mathsf{y}^n_{t,i}\}$. Consequently, the n^{th} particle at time t is represented as $X^{(n)}_t = \{I^n_t, x^n_{(t,i)}\}$, $i \in \{1, ..., I^n_t\}$, where I^n_t is the number of tracked persons by this particle at time t.

6.3.2.2 Proposal Moves

Four sets of proposal moves are used: $m = \{\texttt{Add}, \texttt{Update}, \texttt{Remove}, \texttt{Swap}\}$. The choice of the proposal privileged in each iteration is determined by q_m, the jump move distribution. These values are determined empirically and are set to $\{0.15, 0.8, 0.02, 0.03\}$ respectively. They are tuned to better reflect the occurrences of these events in the scene. It is evident that, once a target appears in the scene, he/she does not disappear immediately. So there will more `Update` moves rather than `Add`, `Remove`, and `Swap` moves. These values could actual be set arbitrary, but then a lot of MCMC iterations would be required to obtain a steady state approximation of

Algorithm 6.1 RJMCMC-PF based Multi-person Tracking Implementation

procedure RJMCMC_TRACK($\{X_{t-1}^{(n)}\}_{n=1}^{N}$; $\hat{X}_{t-1}$; $\{z_t\}$)
  Init: pick a random particle from the set $\{X_{t-1}\}$ with similar configuration to $\hat{X}_{t-1}$ and perturb each target with a zero mean Gaussian to obtain X_t^0;
  for $n \leftarrow 1$ to $N_T N + N_B$ **do**
    Choose a move $m \in \{\texttt{Add}, \texttt{Update}, \texttt{Remove}, \texttt{Swap}\} \sim q_m$
    switch m **do**
      case `Add`:
        $X^* = \{X_t^{(n-1)}, x_p\}$; x_p is randomly taken from $\{z_{t,j}^d\}$
        $\beta = \min\left(1, \frac{p(Z_t|X^*)Q_{remove}\left(X_t^{(n-1)}|X^*,Z_t\right)q_{remove}\Psi(X^*)}{p\left(Z_t|X_t^{(n-1)}\right)Q_{add}\left(X^*|X_t^{(n-1)},Z_t\right)q_{add}\Psi\left(X_t^{n-1}\right)}\right)$
      case `Remove`:
        $X^* = \{X_t^{(n-1)} \setminus x_p\}$ where $p \in \{1, ..., I^{n-1}\}$
        $\beta = \min\left(1, \frac{p(Z_t|X^*)Q_{add}\left(X_t^{(n-1)}|X^*,Z_t\right)q_{add}\Psi(X^*)}{p\left(Z_t|X_t^{(n-1)}\right)Q_{remove}\left(X^*|X_t^{(n-1)},Z_t\right)q_{remove}\Psi\left(X_t^{(n-1)}\right)}\right)$
      case `Update`:
        Randomly select a target x_p from $X_t^{(n-1)}$
        Select x_p^*, a random subspace corresponding to x_p in the particle set at $t-1$
        Replace x_p with a sample from $\mathcal{N}\left(;x_p^*,\Sigma\right)$ proposing X^*
        $\beta = \min\left(1, \frac{p(Z_t|X^*)\Psi(X^*)p\left(X_t^{(n-1)}|X_{t-1}\right)}{p\left(Z_t|X_t^{(n-1)}\right)\Psi\left(X_t^{(n-1)}\right)p(X^*|X_{t-1})}\right)$
      case `Swap`:
        Swap the ids of two near tracked persons to propose X^*
        $\beta = \min\left(1, \frac{p(Z_t|X^*)\Psi(X^*)}{p\left(Z_t|X_t^{(n-1)}\right)\Psi\left(X_t^{(n-1)}\right)}\right)$
    if $\beta \geq 1$ **then**
      $X_t^{(n)} \leftarrow X^*$
    else
      Accept $X_t^{(n)} \leftarrow X^*$ with probability β or reject and set $X_t^{(n)} \leftarrow X_t^{(n-1)}$
    end if
  end for
  · Discard the first N_B samples of the chain and retain only every N_T^{th} particle
  · Identify the mode of the particle configurations and define $\tilde{X}_t$ as containing only those particles conforming with the mode configuration
  · Point estimate, $\hat{X}_t := E_{p(X_t|Z_{1:t})}[\tilde{X}_t]$
  return $\{X_t^{(n)}\}_{n=1}^{N}$ and $\hat{X}_t$
end procedure

the posterior. To formulate the proposal move distributions, $Q_m()$, a Gaussian Mixture model is used. A Gaussian distribution better represents the confidence obtained from a detector and tracker that provides a point estimate for the target position. This distribution clearly exemplifies the highest confidence at the point estimate (mean) and how the confidence wears off as we move away from the centroid radially.

To simplify both the transition of the new proposed state hypothesis X^* (at the n^{th} iteration from $X_t^{(n-1)}$ at time t) and evaluation of the acceptance ratio only changes to a randomly chosen subset of the state is considered. In multi-target tracking, this translates into changing a single target per iteration.

Add: The add move, randomly selects a detected person, x_p, from the pool of provided detections and appends its state vector on $X_t^{(n-1)}$ resulting in a proposal state X^*. The proposal density driving the `Add` proposal, $Q_{Add}\left(X^*|X_t^{(n-1)}, Z_t\right)$, is then computed according to equation 6.2. This equation represents a mixture of Gaussian map made from the detected persons and tracked persons at time $t-1$. Each detection is represented as a Gaussian on the ground plane. It is then masked by a similar mixture derived from the tracked persons (Maximum A Posteriori (MAP) estimate $\hat{X}$) at time $t-1$ in such a way that the distribution will have higher values on locations conforming to detected persons that are not yet being tracked. The covariance matrix used for all Gaussian mixtures from detector and tracking are kept identical to simplify normalization.

$$Q_{add}\left(X^*|X_t^{(n-1)}, Z_t\right) = \left(\sum_d \frac{k_d}{N_d}\sum_{j=1}^{N_d}\mathcal{N}\left(x_p; z_{t,j}^d, \Sigma\right)\right).\left(1 - \frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{N}\left(x_p; \hat{X}_{t-1,j}, \Sigma\right)\right) \tag{6.2}$$

Where d represents the set of detectors, namely: from laser (l), fixed camera 1 (c_1), and fixed camera 2 (c_2); $d \in \{l, c_1, c_2\}$ (each camera has two detections: HOG, hog, and Foreground Segmentation, $fseg$), N_d is the total number of detections in each detector, k_d is a weighting term for each detector such that $\sum_d k_d = 1$, $\hat{X}_{t-1}$ is the MAP estimate of the filter at time $t-1$, and N_T is the number of targets in this MAP. Figure 6.7 clearly illustrates what the add move proposal density looks like on a specific situation. When a new person is added, its appearance is cross-checked with the appearance of targets that have been tracked. If there is a high similarity, determined based on Bhattacharyya distance, the new person is given the id of the matched person and the situation is treated as a simple re-identification step.

Remove: The remove move, randomly selects a tracked person x_p from the particle being considered, $X_t^{(n-1)}$, and removes it, proposing a news state X^*. Contrary to the add move, the proposal density used when computing the acceptance ratio, $Q_{Remove}(X^*|X_t^{(n-1)})$ (equation 6.3), is given by the distribution map from the tracked persons masked by a map derived from the detected persons. This distribution favors removal of targets that have gone out of the tracking area but are still being tracked.

$$Q_{remove}\left(X^*|X_t^{(n-1)}, Z_t\right) = \left(1 - \sum_d \frac{k_d}{N_d}\sum_{j=1}^{N_d}\mathcal{N}\left(x_p; z_{t,j}^d, \Sigma\right)\right).\left(\frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{N}\left(x_p; \hat{X}_{t-1,j}, \Sigma\right)\right) \tag{6.3}$$

Even though the tracker of a person who left the scene ceases to exist, a dynamic appearance model of the person is kept for a later re-identification.

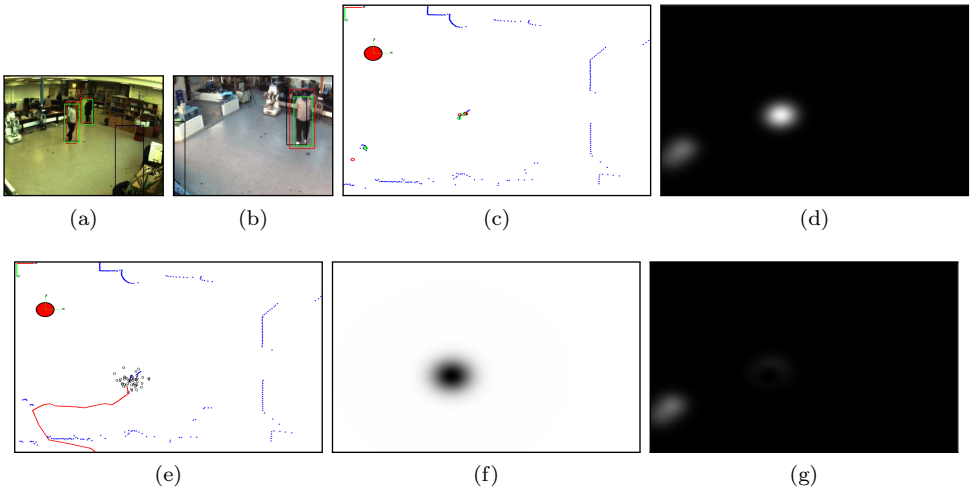


Figure 6.7: Illustration of the add proposal distribution. (a)-(b) shows the wall mounted cameras feed with various detections (laser in black, foreground segmentation in red, and HOG in green). (c) projection of each detection onto the ground plane. (d) shows the mixture of Gaussian distribution determined from the various detections. (e) shows the tracked target at time $t-1$ and (f) shows its corresponding Gaussian mask. Finally, (g) shows the add proposal distribution obtained by masking (d) with (f), and it indeed shows salient values on the position of the untracked person.

Update: In the update proposal move, the state vector of a randomly chosen target is perturbed by a zero mean normal distribution. The update proposal density, $Q_{update}(X^*|X_t^{(n-1)}, Z_t)$, is a normal distribution with the position of the newly updated target as mean. Hence, the acceptance ratio is influenced only by the likelihood evaluation and interaction amongst the targets.

Swap: The swap move handles the possibility of id switches amongst near or interacting targets. When this move is selected, the ids of the two nearest tracked persons are swapped and a new hypothesis X^* is proposed. The acceptance ratio is computed similar to the `Update` move.

6.3.2.3 Interaction Model ($\Psi(.)$)

Since the persons in the surveilled area are likely to interact, an Interaction Model is included to maintain tracked person identity and penalize fitting of two trackers to the same object during interaction as discussed in section 5.2.2. Similar to [Khan 2005, Smith 2005], a Markov Random Field (MRF) is adopted to address this. A pairwise MRF where the cliques are restricted to the pairs of nodes (targets define the nodes of the graph), that are directly connected to the graph, is implemented as part of our tracker. For a given state $X_t^{(n)}$, the MRF model is given by equation 6.4. As can be seen from this equation, as long as the σ term is not set to zero, $\phi(,)$ will always be greater than 0 and less than 1. The sigma determines how well the effect should be pronounced when the targets are close by.

$$\begin{aligned} \Psi(X_t^{(n)}) &= \Pi_{i \neq j} \phi(x_{t,i}^n, x_{t,j}^n) \\ \phi(x_{t,i}^n, x_{t,j}^n) &= 1 - exp\left(-\left(\frac{d(x_{t,i}^n, x_{t,j}^n)}{\sigma}\right)^2\right) \end{aligned} \tag{6.4}$$

where $d(x^n_{t,i}, x^n_{t,j})$ is Euclidean distance; $i, j \in \{1, ..., I^n_t\}$; and I^n_t is the number of targets in X^n_t.

6.3.2.4 Observation Likelihood ($p\left(Z_t|X^{(n)}_t\right)$)

The observation likelihood, $p\left(Z_t|X^{(n)}_t\right)$, is derived from all detector outputs except the LRF for which blobs formed from the raw laser range data are considered. If the specific proposal move is an `Update` or `Swap` move, a Bhattacharyya likelihood measure is also incorporated. The raw laser data is filtered to make blob and keep those within a range of radius, denoted as l_b. This filters out laser data pertaining to walls, thin table or chair legs, and other wide structures. Then every filtered blob is represented as a Gaussian on the ground plane centered on the centroid of the blob. HOG based person detection, and detection from foreground segmentation are also represented as a Gaussian mixtures on the ground plane averaged over the number of detections with each Gaussian centered on the detection points. Representing the measurement information at time t as z_t, the observation likelihood of the n^{th} particle X^n_t at time t is computed as shown in equation 6.5.

$$
\begin{aligned}
p\left(Z_t|X^{(n)}_t\right) &= \pi_B\left(X^{(n)}_t\right).\pi_D\left(X^{(n)}_t\right) \qquad (6.5)\\
\pi_B\left(X^{(n)}_t\right) &= \begin{cases} \prod_{i=1}^{M}\prod_{c=1}^{2} e^{-\lambda B^2_{i,c}}, & \text{if } move = \texttt{Update or Swap} \\ 1 & \text{otherwise} \end{cases}\\
\pi_D\left(X^{(n)}_t\right) &= \frac{1}{M}\sum_{i=1}^{M}\left(\sum_d k_d.\pi\left(x_i|z^d_t\right)\right), \sum_d k_d = 1\\
\pi\left(x_i|z^d_t\right) &= \frac{1}{N_d}\sum_{j=1}^{N_d}\mathcal{N}\left(x_i; z^d_{t,j}, \Sigma\right)
\end{aligned}
$$

In equation 6.5, $B_{i,c}$ represents the Bhattacharyya distance computed between the appearance histogram of a proposed target i in particle X^n_t and the target model in each camera c. M represents the number of targets in the particle, and N_d the total number of detections in each detection modality d, $d = \{l_b, c_1, c_2\}$, in this case including the measures from the laser blobs. k_d is a weight assigned to each detection modality taking their respective accuracy into consideration and x_i represents the position of target i in the ground plane.

At this point, it is interesting to point out that, even though the fusion of information from only three sensors (laser and two wall mounted cameras) is considered, the framework is equally applicable for the fusion of more heterogeneous sensors.

6.3.2.5 Adaptive Color Appearance Model

For each tracked person, an adaptive color appearance model in the form of an HS+V histogram per camera, h^c_{id}, is stored. This histogram is kept even after the targets have left the scene. It is mainly used to re-identify a previously tracked person when a new track is initiated on him/her. The new track could be initiated either due to re-entrance of the person in the surveiled arena once having left, or re-initialization after tracker failure. Whenever a new person is added, its color histogram is cross-checked with existing models. If the Bhattacharyya distance is below a threshold value β_o, the new track is given the id corresponding to the matched histogram. In each time step, the appearance model of tracked persons is updated according to equation 6.6

only if the Bhattacharyya distance with the adaptive model and estimated target histogram is below a threshold value β_t.

$$h^c_{id}(t) = \alpha * h^c_{id}(t-1) + (1-\alpha) * \hat{h}^c_{id}(t) \tag{6.6}$$

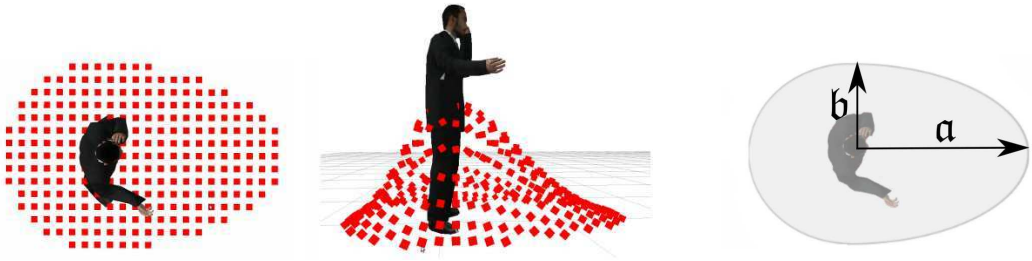
Where $h^c_{id}(t)$ represents the adaptive histogram of target id in the camera c at time step t, and $\hat{h}^c_{id}$ corresponds to the current target's appearance computed at the estimated position. α is a weighting term that determines how much the current appearance affects the global model.

6.4 Robot Navigation Aspects

The mobile robot which makes one part of the cooperative perception framework presented is bound to move in the surveilled environment. Actually, its mobility is one of the motivations that fueled its utilization. Since the environment is co-occupied by people, special care must be taken to realize a safe navigation by the robot. As a result, the mobile robot should take the perceived position and motion direction of the people into consideration when navigating. To this end, we will discuss two points: (1) the personal space model of each tracked individual, and (2) given a goal, a reactive navigation strategy that allows safe robot navigation to goal with dynamic obstacle (people) avoidance (taking the people's personal space into consideration).

6.4.1 Personal Space Model

By definition, a personal space is the area individual humans actively maintain around themselves into which others cannot intrude without arousing discomfort [Hayduk 1978]. Many researchers have proposed different shapes to model the personal space, but most of them agree on asymmetric space which provides more room up front as people in general are more strict regarding their frontal space [Vasquez 2012]. Inspired by the personal space proposed by [Vasquez 2012] which considers blending two Gaussian functions centered at the position of the person, one Gaussian representing front while the other represents the behind space, we use the strict personal space made up of two half ellipses as shown in figure 6.8c. We consider this simplification as our main objective here is to validate the perceptual capabilities to respect a specified constraint and not to investigate actual social cognition during navigation.



(a) (b) (c)

Figure 6.8: Personal space models. (a) and (b) show the model based on Gaussian functions used in [Vasquez 2012], and (c) shows a simplified elliptical discrete zone model.

6.4.2 Nearness Diagram (ND) Navigation

The Nearness Diagram (ND) navigation is a navigation algorithm which is based on identification of free space areas and obstacles proximity based on some diagrams to define a set of situation which trigger specific motion laws. A situation is identified by the pose of the robot, the obstacle distribution, and the goal location. All known situations are used to build a decision tree beforehand. Then for any situation encountered, the tree is traversed based on binary decision rules that evaluate the current situation resulting (identifying) an associated action (control law) to use for this scenario. This specific obstacle avoidance has been chosen owing to its simplicity, real-time performance, and as it has been demonstrated to be an effective navigation method capable of avoiding collision in troublesome scenarios [Minguez 2004]. In our case, the only modification is: instead of seeing persons as just static obstacles, the algorithm will treat them as obstacles with special zone needs that depend on their motion direction, the special zone depicted in figure 6.8c.

6.5 Evaluations and Results

In this section the off-line and on-line (live) experiments carried out to evaluate the performance of the implemented multi-person tracker are presented. For the off-line evaluation, a set of datasets are acquired and the multi-person tracker is evaluated on them afterwards. For the on-line evaluation, all tracking and metric computations are performed as the experiment progresses live.

6.5.1 Off-line Evaluation

To evaluate the performance of our RJMCMC-PF multi-person tracker, three sequences acquired using Rackham (kept static during acquisition) and the wall mounted cameras are used. The sequences are acquired inside LAAS's robotic room which has an area of approximately $8 \times 6m^2$ where Rackham can actually move. Each sequence contains a laser scan and video stream from both cameras. Sequence I is a 200 frame sequence containing two targets in each frame. Similarly, sequence II contains 200 frames featuring three moving targets. Sequence III contains four targets moving in the vicinity of the robot and is 186 frames long. The quantitative performance evaluation is carried out using the evaluation criteria presented in section 5.2.4 (which are actually based on the CLEAR MOT metrics [Bernardin 2008]) and presented in section 5.2.4.

For evaluation, a hand labeled ground truth with (x, y) ground position and unique id for each person is used for each sequence. A person is considered to be correctly tracked (True Success), if the tracking position is within a 30 cm radius of the ground truth. For people detector, Dalal and Triggs HOG [Dalal 2005] is used. But, given this is an off-line evaluation and our proposed detectors, HOG-BIPBoost and HETERO-BIPBoost, exhibit comparable detection performance to Dalal and Triggs HOG (see figure 4.17 in section 4.17), it is safe to assume the presented results would hold. Each sequence is run eight times to account for the stochastic nature of the filter. Results are reported as mean value and associated standard deviation. The values set for various parameters (determined empirically) to produce the evaluation results reported in this section are listed in table 6.1.

To clearly highlight the advantage of using each sensor, the multi-person tracker is evaluated based on the following different modes:

1. Multi-person tracking using LRF input only. Results are reported in table 6.2. In this case, a detector weight of 1.0 is used for the laser and zero for the rest.

Table 6.1: Parameter values used to produce the results reported in this section.

Symbol	Stands for	Value
k_d	detector weights, $d = \{l, c_1, c_2\}$ with $c_i = \{fseg_{ci}, hog_{ci}\}$	$k_d = \{0.16, \{0.22, 0.2\}, \{0.22, 0.2\}\}$
q_m	jump move distribution	$q_m = \{0.15, 0.8, 0.02, 0.03\}$
Σ	random walk and Gaussian mixture covariance (m^2 units)	$\begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}$
σ	interaction model standard deviation (cm)	75 cms
N	number of particles in RJMCMC-PF	150
N_B	number of burn-in iterations in RJMCMC-PF	40
N_T	number of thin-out iterations in RJMCMC-PF	1
HS+V bins	color histogram bins	8×8 HS bin, 8 V bin
β_t	passer-by appearance model update threshold	0.24
β_o	threshold for conclusive similarity of a new passer-by with existing color model	0.1
α	passers-by dynamic color model update weight	0.9

2. Multi-person tracking using the wall mounted cameras only. Similarly, results are reported in table 6.3. A detector weight of 0.5 is used for each camera with equally influential HOG and foreground segmentation detections and zero for the laser.

3. Cooperative multi-person tracking using a single camera and LRF. The results pertaining to this evaluation mode are reported in table 6.4. The corresponding detector weight used is a 0.4 for the laser and a 0.6 for the camera.

4. Complete system, multi-person tracking using the two wall-mounted cameras and LRF. Results are reported in 6.5. The detector weight parameters reported in table 6.1 are used.

Table 6.2: Laser-based only perception.

Sequence	TSR		MR		GR		Mismatch		**MOTP**		**MOTA**	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
I	0.757	0.034	0.252	0.034	0.396	0.042	15.00	2.618	15.62	2.340	0.410	0.049
II	0.667	0.033	0.333	0.033	0.527	0.104	21.62	5.450	19.90	1.664	0.273	0.068
III	0.606	0.044	0.394	0.044	0.541	0.103	46.75	4.921	21.94	1.745	0.202	0.068

Table 6.3: Wall-mounted cameras-based only perception.

Sequence	TSR		MR		GR		Mismatch		**MOTP**		**MOTA**	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
I	0.897	0.006	0.103	0.006	0.087	0.034	7.60	1.817	19.80	0.140	0.797	0.025
II	0.817	0.049	0.182	0.048	0.089	0.017	19.17	3.920	22.79	1.350	0.708	0.05
III	0.734	0.050	0.265	0.050	0.248	0.016	57.60	14.15	28.44	1.601	0.4588	0.067

The results presented from table 6.2 to table 6.6 clearly attest the improvements in perception brought by the cooperative fusion of LRF and wall mounted camera percepts. The cooperative system consisting of LRF and two wall mounted cameras exhibit an MOTA of 0.841 when tracking two targets, 0.793 for three targets, and 0.538 for four targets with a 93.4%, 88.5%, and 75.5% True Success Rates respectively. The worst average precision is less than 22 *cms*. These results

Table 6.4: Cooperative perception using a single wall-mounted camera.

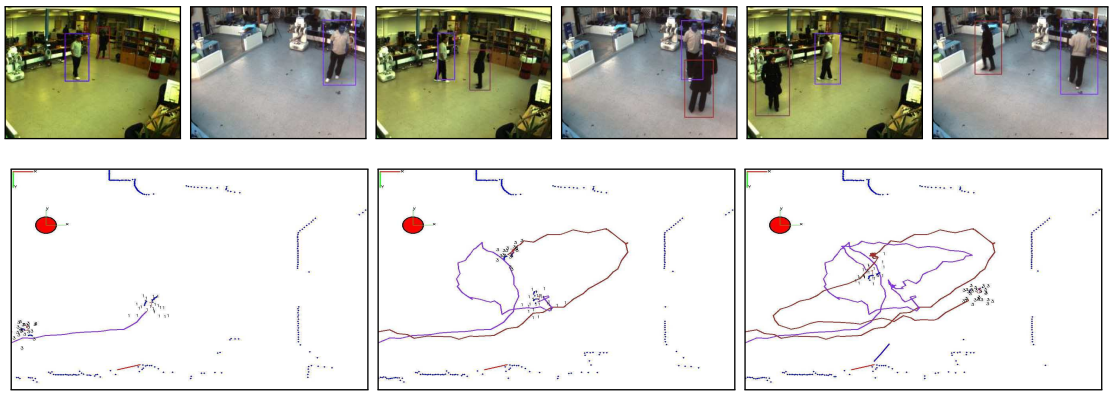
Sequence	TSR		MR		GR		Mismatch		**MOTP**		**MOTA**	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
I	0.932	0.023	0.068	0.023	0.110	0.014	1.333	1.633	17.52	1.80	0.825	0.030
II	0.859	0.032	0.140	0.032	0.147	0.030	10.50	4.680	17.63	1.643	0.713	0.055
III	0.725	0.037	0.274	0.037	0.339	0.069	47.40	6.986	22.83	1.00	0.402	0.051

Table 6.5: Cooperative perception using the two wall-mounted cameras.

Sequence	TSR		MR		GR		Mismatch		**MOTP**		**MOTA**	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
I	0.935	0.029	0.065	0.022	0.099	0.020	0.667	0.816	17.01	1.886	0.841	0.033
II	0.885	0.029	0.115	0.029	0.099	0.020	11.40	3.782	17.73	0.005	0.793	0.030
III	0.755	0.018	0.245	0.018	0.211	0.027	35.60	5.941	21.30	1.358	0.538	0.040

Table 6.6: Id swap occurrences in each tracking mode.

Sequence	LRF-only		Wall-mounted cameras		Cooperative Perception			
					Single Camera		Two Cameras	
	μ	σ	μ	σ	μ	σ	μ	σ
I	2.50	0.76	0.60	0.55	0.00	0.00	**0.00**	**0.00**
II	4.62	0.74	1.33	0.52	0.83	0.41	**0.40**	**0.55**
III	4.88	1.35	2.40	0.55	1.60	0.89	**1.20**	**1.09**

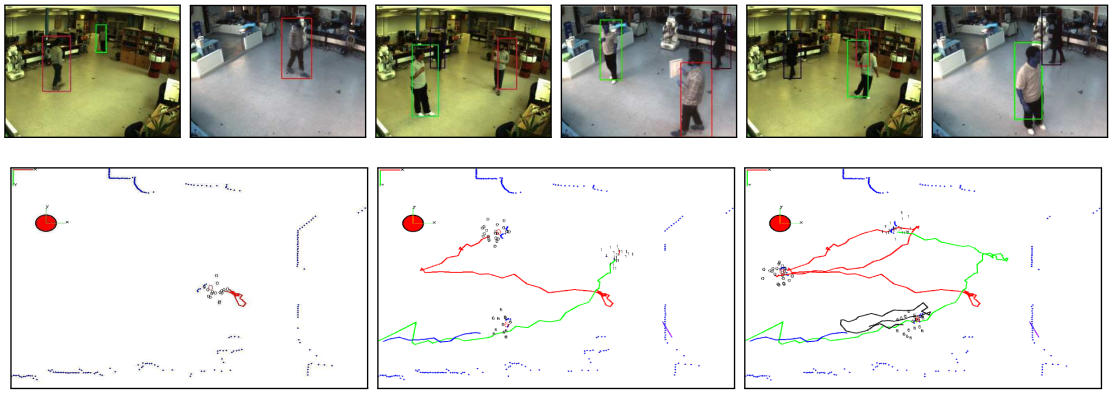


(a) (b) (c)

Figure 6.9: Multi-person tracking illustrations taken from sequence I at a) frame 31, b) frame 80, and c) frame 148. The top two images correspond to the camera streams and the bottom one shows the ground floor with trajectories of tracked persons superimposed. The particle swarm is also shown with the ID of each individual. The small blue dots are the raw laser scan points.

are clearly indicative of how well the system does. Sample tracking sequences for two targets and three targets are shown in figures 6.9 and 6.10 consecutively [1]. Another main observation to make is the low accuracy of tracking based on LRF only. The mistakes made with leg like structures in the environment, sensitivity to occlusion, and lack of discriminating information amongst tracked passers-by corroborate to its poor performance. The results obtained using the wall mounted cameras show major improvements though their position tracking precision is relatively lower compared to those which include LRF measurement. By comparing table 6.4 and

[1] Please visit the URL homepages.laas.fr/aamekonn/phd_thesis/ for complete runs.



(a) (b) (c)

Figure 6.10: Multi-person tracking illustrations taken from sequence II at a) frame 27, b) frame 60, and c) frame 94. The top two images correspond to the camera streams and the bottom one shows the ground floor with trajectories of tracked persons superimposed.

6.5, it is possible to observe that the addition of the second camera in the cooperative scheme improves the tracking results further.

In table 6.6, id swap occurrences under each mode are reported. As specified earlier, this quantity is important to our system because identifier swaps would lead to a false motion estimation which in turn would affect the navigation of the mobile robot. Amongst the reported modes, LRF only tracking does worst. This is again expected as no appearance information to identify one person from another. Hence, LRFs should be used in conjunction with another sensor that has discriminating information where ever possible. Again, the cooperative system with two cameras results in the best results, with almost no id swaps when tracking two and three targets, and 1 to 2 id swaps with four targets through out the sequence.

Two main conclusions can generally be drawn from the results reported in this section. First, classical video-surveillance approaches that rely on fixed visual sensors improved the perception capability of a mobile sensor unit (in our case a robot). The improvement, which was certainly expected, clearly comes from the cameras that provide rich global wide field of view feed to the robot. For the second case, lets consider the evaluation that uses only deported cameras. This system is basically the same as a typical video-surveillance system made up of two networked cameras. The algorithms that we have proposed and implemented are variants of currently considered state-of-the-art algorithms. But, these results were further improved by the addition of a mobile sensor unit. Hence, it can be claimed indoor video-surveillance systems can be generally improved with a mobile sensor unit which on top of everything is also a means for action.

In short, even though the actual reported results depend on the used environment structure, it is clear that the fusion of heterogeneous sensors cooperatively increases the performance consistently. On another note, the implemented passers-by tracking has some pitfalls. Its first shortcoming comes from a formulation inherent in the RJMCMC-PF. The interaction model in this tracker depends on the state-space of the particles and not on the observations. It relies on the inference rather than the evidence. The second shortcoming relates to the employed simple persons' appearance model. Whenever a track fails (loses its target), a new track is initialized after cross-checking the appearance with past tracked targets. If this appearance is not very discriminative, it could lead to a new track initialization rather than assigning the lost track to the current target. Briefly, the simple histogram based appearance model used could easily

confuse persons with similar clothing and lead to erroneous interpretations.

6.5.2 On-line Evaluation

In this section, all on-line evaluation carried out along with obtained results are discussed. The term on-line is used here to mean live runs performed after complete developed system integration. In this sense, we have carried out two types of on-line evaluations.

- Tracker evaluation: where the multi-person tracker is evaluated via live runs using ground truth acquired from a motion capture system. The multi-person tracker evaluation metrics used in section 6.5.1 are again used.
- Safe robotic navigation: in this evaluation, the robot is made to navigate from one starting point to an end point. On each run, people are made to interfere with it by crossing its motion direction. If the robot manages to adjust its path without violating the security zone around each person, then the mission is a success, otherwise it is a failure. This success/failure rate is quantified over several runs.

In both cases, the evaluations are carried out using Dalal and Triggs [Dalal 2005] detector variant and our HOG-BIPBoost implementation variant. These evaluations will also highlight the impact the detector frame rate has on the entire system functioning.

Table 6.7: Computation time [a] taken by the different components of our multi-person tracker.

Function	Average Computation Time (ms)
Data acquisition	50 ms
RJMCMC-PF tracking	100 ms
Background subtraction	10 ms
Leg detection (from laser)	3 ms
Visualization and data serialization	20 ms
Total	183 ms

[a] on an Intel(R) Core(TM) i7-2720QM CPU @ 2.40GHz machine

Before going directly into evaluation, lets analyze the computation time associated with the perceptual function—multi-person detection and tracking. If we forget about the people detector component, the average time taken by the different components on our machine is shown in table 6.7. This shows, on average, the multi-person perception module discarding the detector component takes around 183 milliseconds. Now, coming back to the people detector variants, we have various choices from the ones presented in part I of this manuscript. We have also considered GPU implementations (secion 3.8.3). Hence, we have: the classical HOG from Dalal and Triggs (Dalal and Triggs HOG) and its GPU implementation (GPU-HOG); our HOG-BIPBoost and its GPU implementation (GPU-HOG-BIPBoost); and the detectors based on heterogeneous features with BIP (HETERO-BIPBoost) and its GPU variant (GPU-HETERO-BIPBoost). The average frame rates achieved by these detectors alone and integrated with the tracking modules are shown in table 6.8. Bear in mind, with the tracking, the detectors have to be run on two images (an image from each wall mounted camera). The HETERO-BIPBoost and GPU-HETERO-BIPBoost variants are shown shaded in table 6.8 as their implementations have

not yet been finalized. The reported frame rates are estimated using the insight in the actual frame rate observed from the other detectors. With BIPBoost variants, the models trained with the Ladybug dataset are considered.

Table 6.8: Frame rate [a] [b] achieved by the various detectors, alone and integrated with the tracking framework.

Detector	Average fps	With Tracking
Dalal and Triggs HOG	0.65 fps	0.314 fps
HOG-BIPBoost	3.0 fps	1.17 fps
GPU-HOG	7.7 fps	2.26 fps
GPU-HOG-BIPBoost	12.5 fps	2.92 fps
HETERO-BIPBoost	≈ 10.0 fps	2.61 fps
GPU-HETERO-BIPBoost	≈ 16.0 fps	3.25 fps

[a] CPU: an Intel(R) Core(TM) i7-2720QM CPU @ 2.40GHz machine
[b] GPU: an nVidia GeForce GF108 (Quadro 1000M)

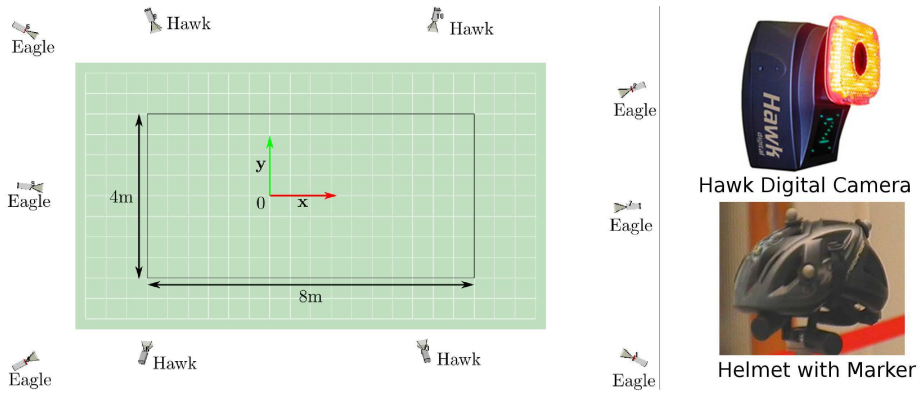


Figure 6.11: Motion capture cameras (from Motion Analysis) and rough orientation on the experimental area (left). An illustrative actual motion capture camera and the helmet tagged with motion capture tags worn by targets for accurate target localization (right).

6.5.2.1 Tracker Evaluation

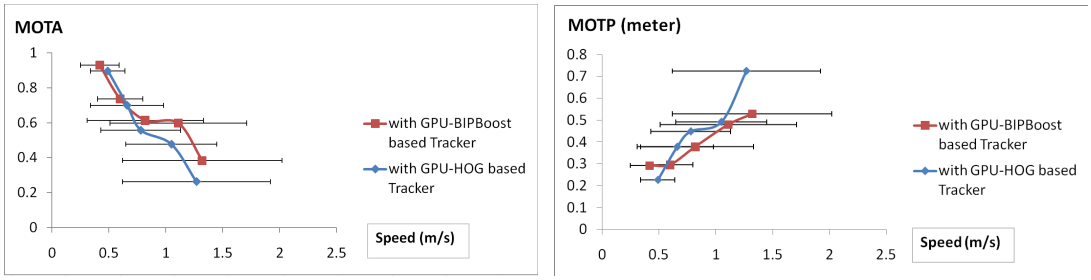
To evaluate the multi-person tracker on-line, all implemented components have been integrated with the platform depicted in section 6.1. Then each target is fitted with reflective markers attached on a helmet (figure 6.11 right). The markers are tracked by the motion capture system deployed in our laboratory (figure 6.11 shows the motion capture IR cameras and their orientation towards the open zone monitored by the proposed cooperative system) and provides millimeteric accuracy. It also keeps identity of the marked targets consistently. In the evaluation, the multi-person tracking parameters presented in table 6.1 are again used. The evaluation alternates between the GPU-HOG and the GPU-HOG-BIPBoost detector variants in consecutive runs. Two targets are used mostly for the evaluation. Corresponding results are shown in tables 6.9 and 6.10. The results are averaged over several runs comprising a total of more than 2600 frames for each tracker using the different detectors. The targets used for the experiments were from the robotic/computer vision domain and they were told to just walk in casual manner.

Table 6.9: Cooperative perception on-line evaluation.

People Tracker	TSR		MR		GR		ID Swap		Mismatch	
	μ	σ	μ	σ	μ	σ	μ	σ		
with GPU-HOG	0.85	0.04	0.10	0.07	0.16	0.07	2.50	1.05	5.17	2.13
with GPU-HOG-BIPBoost	0.89	0.03	0.13	0.08	0.12	0.04	2.33	1.50	4.17	2.13

Table 6.10: Cooperative perception on-line evaluation (MOTP and MOTA).

People Tracker	**MOTP**		**MOTA**	
	μ	σ	μ	σ
with GPU-HOG	24.30	5.40	0.72	0.10
with GPU-HOG-BIPBoost	22.40	6.00	0.74	0.09



(a) (b)

Figure 6.12: Multi-person tracking evaluation with variable walking speed of targets.

As can be seen, the results are quite similar to the off-line results obtained using sequence II. In general, the perceptual modalities using the two detectors achieve more than 85% percent true success rate with a tracking precision of a little over 20 cms with an accuracy of 72.0% when using GPU-HOG and 74.0% when using our proposed GPU-HOG-BIPBoost detector. Under each evaluation criteria, the multi-person tracker based on the GPU-HOG-BIPBoost excels. Illustrative videos corresponding to a few of these runs are available at the URL homepages.laas.fr/aamekonn/phd_thesis/.

To see the effect people's walking speed has on the tracker (which rely on detectors with different frame rates), we carried out the following set of experiments. The targets were asked to walk at different speeds during each experiment run. The $30cm$ constraint for ground truth association is relaxed to 1.0 meter. Then, multiple runs are evaluated using the GPU-HOG and GPU-HOG-BIPBoost detector variants alternatively. Figure 6.12a shows the resulting MOTA as a function of the targets speed and figure 6.12b that of MOTP. For targets' speed, what we have done is, to measure their speed (obtained using the motion capture system) at each frame and then determine the average speed and its standard deviation using the data of only a complete run. We make sure each run is more than 200 frames to gather characteristic evaluation. In the figures 6.12a and 6.12b, a marked point corresponds to a single trial with its x-value corresponding to the average speed and y-value to the measured quantity. The standard deviation shown (in the form of error margin) provides intuitive information how the targets' motion varied throughout the run.

The results obtained in both figures (6.12a and 6.12b) unanimously show, as the motion speed of the targets increases, the performance difference between the two tracker modes (based on GPU-HOG and GPU-HOG-BIPBoost) widens. For example, at an approximate average speed of 1.25 m/s, the GPU-HOG-BIPBoost based tracker shows a close to 20% gain in MOTA, compared to the less than 5% gain at around 0.5 m/s. The gap in tracker precision also widens similarly. The GPU-HOG-BIPBoost variant shows a 27.8% precision improvement over the GPU-HOG based variant at an approximate average speed of 1.25 m/s. Hence, looking at these two graphs, we can conclude that the frame rate improvement of this detector does actually improve the trackers' performance well (recall that the two detectors have comparable detection performance) which becomes apparent as the dynamics of the targets increases.

6.5.2.2 Safe Robotic Navigation

The next experiment involves testing the robot's safe navigation functionalities during robotic mission execution. Based on the different envisaged detectors combined with the tracking, table 6.8 showed the average frame rate achieved on the current system. Using this information, figure 6.13 shows how far the robot needs to be from a person's security zone, to detect the person at the next perception cycle and eventually circumnavigate without violating his/her personal space when traveling at a certain speed (assuming ideal tracking and control cases). For example, when using classical Dalal and Triggs HOG and traveling at 0.3 m/s, the periphery of the security zone of the person will have to be more than 1m far from the robot when the person appears in the FOV, otherwise if it is less than that, the robot would have already violated the region before the next perception comes. Under the same conditions, GPU-HOG-BIPBoosting+Tracking would stipulate less than 10 cms of distance. This increases the robots reactivity, reacts to a personal space just 10 cm further contrary to the 1m requirement. Notice from this graph that as the frame rate increases, the bottleneck slowly shifts towards the other perceptual functionalities only resulting in marginal overall difference. But, this does not mean the obtained frame rate gains are useless. First, even a slight increase in frame rate can have real observable significance during robotic action (see table 6.11). Second, if any speed improvement/boosting is to be achieved with the other components, the speed improvement brought by the detector will be unleashed further improving the overall perceptual frame rate.

To test the safe navigation aspect of our mobile robot using the perceptual input from the cooperative perception, we devised the experiments shown in figure 6.14. In the first kind, figure 6.14a, the robot is made to go from a start point to an end point with a maximum velocity of 0.3 m/s. Eventually, a person traverses perpendicular to the robot motion direction. Then the behavior of the robot's motion is observed. If at any time, the robot crosses/violates the personal space of the person, the mission is considered as a failure. Otherwise, it is counted as a success. The trajectories traversed by the robot and person during the multiple runs are shown in figure 6.14a in red and green respectively. In the second kind, figure 6.14b, the robot is again made to traverse from the start to the end position under similar conditions but this time two people interfered with its motion. The motion of the first person is kept somehow longitudinal to the robot (but in opposite direction) and that of the second person is kept perpendicular. The trajectories of the people and robot are shown in figure 6.14b in blue and green for the people and in red for the robot. Again a mission is counted as successful if none of the person spaces are violated. In both scenarios, the trajectories are shown on the map of the environment constructed from laser segments and which is used by the robot for localization. The ground truth data are acquired using the motion capture system with the help of special reflective markers attached to the robot and people's helmets (figure 6.11). Given the fact that our experimental environment is a bit narrow, we have used a personal space with a major elliptical axis, $\mathfrak{a}$, of 1.0 m and minor

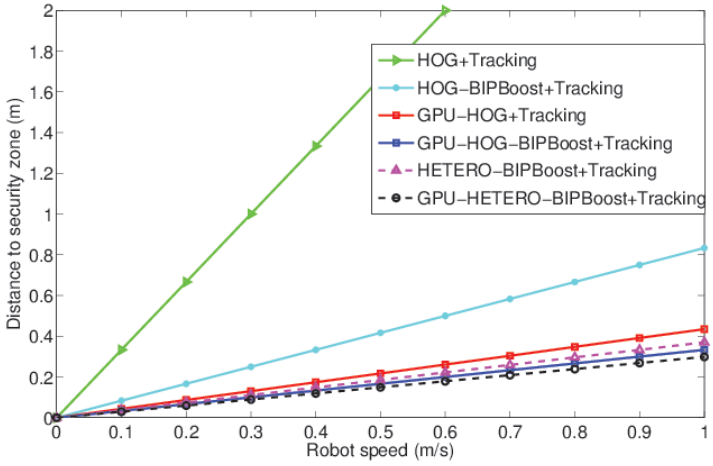


Figure 6.13: The minimum distance between a mobile robot and a person's security zone as a function of the robot's speed necessary to guarantee safe navigation when utilizing the perception mode under the different detectors. The dotted plots are actually based on estimated detector speed rather than actual measurements.

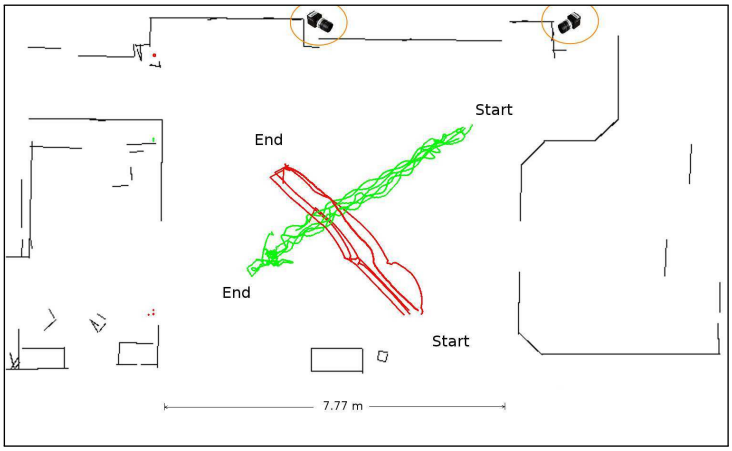
elliptical axis, $\mathfrak{b}$, of 0.6 m (in reference to figure 6.8c). In exactly half of these experiments, the GPU-HOG is used as people detector, and for the other half, our GPU-HOG-BIPBoost is used. Table 6.11 summarizes the results obtained under the two detector variants. The main perceptual inputs to the navigation components are the position and orientation of the tracked targets. The position is straight forward, but for the orientation we use a 3-point moving average filter to smooth orientation angles using the previous two orientations determined at the previous time frame and the one before that. The navigation module first places the specified elliptic personal space at the specified position and along the specified orientation on the raw laser inputs. It then uses the ND navigation scheme, section 6.4.2, to implement a straight forward navigation control using the modified laser data inputs.

Table 6.11: Robotic mission success.

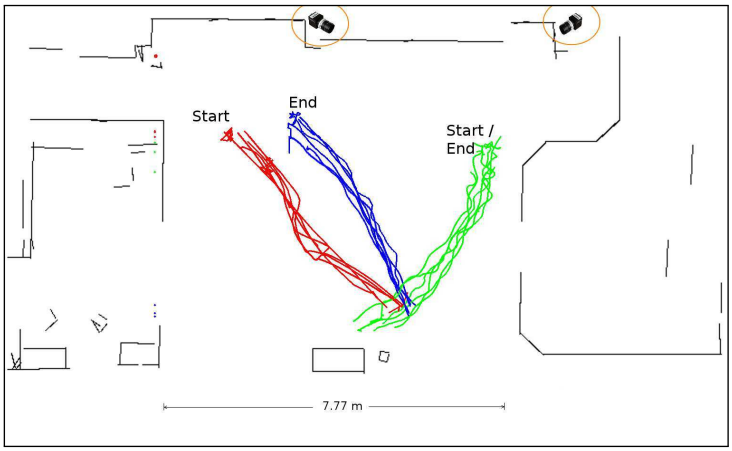
Detector used with Tracker	Average fps	Minimum Distance (m)	Average minimum distance	Mission success rate
GPU-HOG	2.26 fps	0.672 m	1.012 m	66.7%
GPU-HOG-BIPBoost	2.92 fps	0.744 m	1.093 m	83.0%

Table 6.11 summarizes the results obtained during the navigation aspect evaluation. When using the cooperative perception based on our developed GPU-HOG-BIPBoost detector variant, the robot managed to complete missions successfully in 83% of the runs. The minimum distance encountered to a person is 0.744 m and the average minimum distance to a person is 1.093 m. All these values are superior to the alternative variant based on the GPU-HOG which achieves a 66.7% success rate with an encountered minimum distance to a person of 0.672 m. These results attest that even the 0.66 fps incremental frame rate exhibited by the GPU-HOG-BIPBoost does have an impact on the robot reactivity.

Figures 6.15 and 6.16 show sample runs that illustrate a failed and successful robotic mission



(a)



(b)

Figure 6.14: Experiments carried out to evaluate the robot's safe navigation that relies on the developed perceptual inputs. In (a), only a single person interfered with the robot's motion, the green line shows the trajectory traversed by the person and the red that of the robot in multiple runs. In (b), two people interfered with the robot's motion, the blue and green show the trajectories of the people whereas the red shows that of the robot. In all cases, the 'start' and 'end' markers denote the starting and ending positions. If both are used, it means the start and end positions were reversed in half of the runs. All plots are shown on the map of the environment constructed from laser scan segments which are actually used by the robot for localization.

respectively. The perceptual component during the run shown in figure 6.15 uses GPU-HOG as a detector. In the figure, the feed from the cameras are shown on the top row and a birds eye view of the ground plane is shown at the bottom row of each sub figure. The robot with its orientation is shown in red circle. All tracked targets are shown with a colored circular outline with a sticking out orientation vector on the ground plane. The blue dots show the default laser scanned points whereas the black dots show the modifications made to the laser scan based on the perceived position and orientation of tracked targets which governs the pose of the elliptic personal space. Only partial views, sides visible from the laser range finder, of the elliptic personal

zones are shown. The first close encounter between the robot and a person occurs in the 33^{rd} frame (figure 6.15c). Then the rotation of the robot to avoid the personal space can be observed from frames 34 to 36 but not after it violates the person's personal space in frames 35 and 36 which deem the mission as a failure. But, after that the robot correctly manages to avoid the second passer-by which is in the way by respecting its personal space and turning and heading to its goal behind him.

Similarly in figure 6.16 sample snap shots taken from a run which uses the GPU-HOG-BIPBoost detector is shown. In this run, the robot actually manages to finish the mission successfully without violating anyone's personal space. The robot actually starts rotating (for avoidance) just before it violates the personal space of the nearby person in frames 36 to 39. It then again poses a bit and does the same avoidance with the second target reaching its goal successfully. Videos showing the complete runs of figures 6.15 and 6.16 are available at homepages.laas.fr/aamekonn/phd_thesis/

6.6 Towards a Self-Contained Robotic Perceptual System

The cooperative perceptual system, between wall mounted cameras and a mobile robot, presented in this chapter is well adapted for moderate size areas that require increased accuracy. The centralized nature of the employed data fusion makes extension to big environments with multiple cameras infeasible due to huge bandwidth and computation resources requirements. In section 1.1, we presented an exemplar airport scenario that would require different modes to realize surveillance of the entire infrastructure. One possible mode mentioned is a mobile robot with sensors, considered as a mobile sensor unit, that can potentially be used to cover dead-spots in the deployed environment fixed sensor network FOVs, and eventually to decrease the required number of sensors deployed to achieve sufficient coverage. In this section, we will briefly present a work in progress to address this mode.

The envisaged perceptual system uses sensors fully on-boarded on our mobile robot. It is based on a spherical camera, named *Ladybug2*, and laser range finder. The employed framework is similar to the one presented in section 6.2.2, but instead of taking images from the wall mounted cameras, this time they are used from the *Ladybug2* camera. The framework is illustrated in figure 6.17. The firs sensor used, the *Ladybug2*, is not a conventional camera per say. It is actually a camera system composed of six cameras mounted in such a way to view more than 75% of a full sphere. Each camera has a maximum resolution of 1024x768 pixels resulting in a 3500x1750 pixels stitched high resolution panoramic image. The camera system has an IEEE-1394b (FireWire 800) interface that allows streaming at 30 fps with the drivers provided by the manufacturer (only for a windows operating system). The customized drivers developed in situ (in our laboratory) using generic linux support libraries falls short of that achieving at most 10 fps. The acquisition works by first acquiring individual frames at a time from each cameras, and then stitching the images to obtain the stitched omnidirectional images.

The stitching process uses a calibrated polygon mesh data provided by the manufacturer that dictates how the image textures are mapped onto polygons whose geometric vertices are arranged in 3-dimensional coordinate system. This data also comes with alpha values defined per each pixel to directly blend overlapping regions between images from neighboring cameras. Finally, we project the stitched spherical mesh onto a cylinder and unroll it to obtain the omnidirectional image in a conventional format. All these operations are performed using the graphics card via OpenGl interface which relieves significant CPU resources. Once all this is done, a people detector is used to detect all persons around the robot. Since the image is of high resolution, applying conventional single machine detectors would not get us any where. This is in fact one

of the motivations to invest the time and energy to develop the detectors presented in chapters 3 and 4. The GPU-HOG-BIPBoost detectors trained on the Ladybug2 dataset detects people at 7.5 fps on a 1200×386 version of the stitched images.

The second mode of detection is obtained by using the laser range finder. Using the leg detection scheme to detect people, presented in section 6.3.1.1, all people within the 180^o FOV of the laser range finder are independently detected. These detection are projected onto the *Ladybug2* camera images thanks to a calibrated system. The projection yields a bounding box on the image plane taking the height of an average person, $1.7m$, as a standard. These detection are then passed onto the multi-person tracking module.

The multi-person tracking module takes both detections and the associated data (image and raw laser data) as input and tracks the multiple people in the FOV using the RJMCMC-PF. The tracking here is actually performed in the image plane as it provides a complete coverage of the scene with rich discriminant information. The tracking information is eventually used by the mobile robot to realize a navigation scheme that respects the personal space of individuals. Evaluation of the complete system for multi-person tracking and its consequence on the reactivity of the robot during navigation are ongoing.

6.7 Discussions

In this chapter, the primary focus has been on presenting the implementation details of a cooperative framework between wall mounted cameras and sensors on a mobile robot for the detection and tracking of multiple people in a monitored environment. The presented framework follows a centralized cooperation in that all sensor inputs are gathered onto central processor which does the data fusion and tracking. The illustrated framework actually uses two fixed-view cameras mounted on a wall and a laser range finder mounted on a mobile robot. Though, the presentation has been limited to these sensors, it can be extended to accommodate other sensors so long as the data communication bandwidth permits it. The perceptual components are based on three different modes of detection. Background subtraction and a people detector on the wall mounted camera inputs and leg detection on the laser data. The multi-person tracking is realized via RJMCMC-PF which has been shown to be the best alternative when tracking a variable number of targets that are likely to interact.

The first set of evaluations carried out were aimed at validating the adopted cooperative configuration. In that sense, three different datasets were acquired and evaluated off-line. In the evaluations, a laser only system, a laser and a single camera only system, a two camera system, and the complete cooperative system composed of two cameras and a laser, were separately considered. The results, as hypothesized, demonstrated the complete cooperation system outperforms all the other variants in terms of both tracking accuracy (in all three sequences) and tracking precision (in two of the sequences). The complete cooperative system was also less susceptible to id swaps.

The second part of the evaluation has focused on characterizing the on-line (live) performance of the tracker while using different detectors. The comparisons have focused on the GPU versions of the classical Dalal and Triggs HOG detector and our HOG-BIPBoost detector presented in part I of the manuscript. The difference in frame rate between the trackers using the two variants is 0.66 fps (2.26 fps with GPU-HOG and 2.92 fps with GPU-HOG-BIPBoost). On a casual run, the difference between the two was only marginal with the tracker based on GPU-HOG-BIPBoost showing a 2% improvement in tracking accuracy. But, the actual performance difference became when evaluating the trackers under different target speeds. First thing, the performance of both trackers deteriorates with increased target speed. This is expected, especially given the actual low

frame rate (only 2.92 fps for our fast tracker), abrupt motions are likely to harm the system. At the same time, the performance gap between the two trackers starts to widen as the targets speed increases. In fact, at approximately 1.25 m/s average targets speed, the GPU-HOG-BIPBoost based tracker shows a 20% increase in MOTA and a 27.8% improved precision, MOTP, over the tracker based on GPU-HOG. These results affirm the obvious expectation that the slightest improvement in tracker frame rate (under the same conditions, this would translate into the difference in detector frame rate), would lead to improved perception. Hence, it can be boldly stated that the tracker based on GPU-HETERO-BIPBoost (which in our current estimate will achieve an average 3.25 fps) will lead to further perception improvements. On a platform with no GPU support, the non-GPU variants HOG-BIPBoost and HETERO-BIPBoost could be used with the tracker leading to 1.17 fps and 2.61 fps (estimated value). These can still be used as a tracker whereas a tracker based on the non-GPU HOG version will be unusable with a 0.314 fps. In our discussion here, detector detection performance has not been raised because all the considered detectors have comparable detection performance (section 4.6.2).

The last set of evaluations focused on testing the navigation aspect of the mobile robot in people occupied environments. In this vein, the multi-person tracker outputs are used to provide position and orientation of each tracked target in the environment. For each target, we then create an elliptic personal space around each person. With this modification we use a proven dynamic navigation scheme based Nearness Diagram (ND) to realize people avoidance. In the experiments, the robot is instructed to go from a starting position to an end position. In between, people interfered by crossing its direction of motion. If at any moment, the robot violates the personal space of a person, the mission is considered as a failure and otherwise it is a success. Multiple runs were carried out using the multi-person trackers based on GPU-HOG and GPU-HOG-BIPBoost detectors. In our experiments, the robot successfully managed to implement safe navigation with each encountered person 83% of the times with the GPU-HOG-BIPBoost based tracker variant and 66.7% of the times with the GPU-HOG based tracker variant. This demonstrates frame rate improvement brought by our detector does increase the reactivity of the robot. These set of experiments are supposed to be demonstrative and not exhaustive. Our focus in this work is on the perception aspect. We wanted to demonstrat these perception capabilities by endowing the robot with one of its requirements (safe navigation in crowded environments) using ND based navigation. But, our perception and control integration is done in an ad-hoc fashion. In the future, we envisage to investigate well formulated visual servoing techniques, *e.g.*, [Cadenat 2012], for coupling the perception and control aspects in a smooth, coherent, and well formulated fashion.

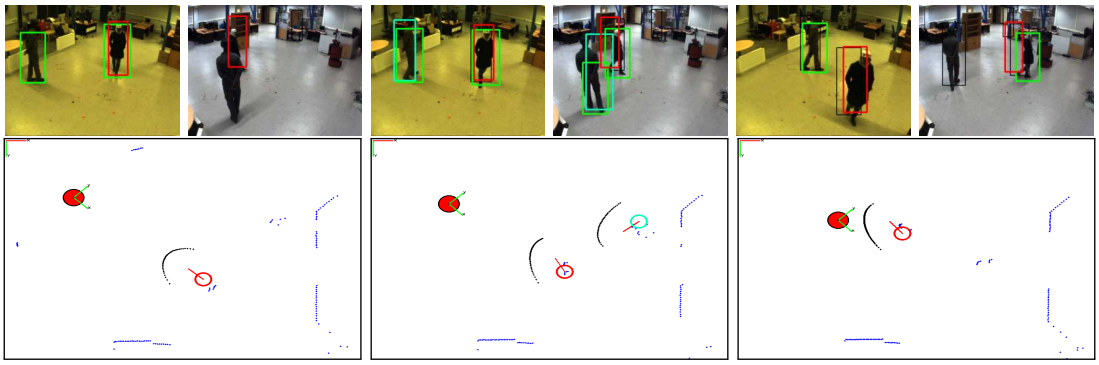
Finally, an excerpt of an ongoing development towards a self-contained robotic perceptual system is presented. The perceptual modalities of this system depend on an omnidirectional camera and a laser range finder. This system is intended to be used as a mobile sensor unit to cover parts of a scene not covered by the deported cameras/sensors. This system is envisaged under the same framework but with sensors mounted on the same platform. Evaluation details of the finalized system will be presented during the Ph.D. defense.

6.8 Conclusion

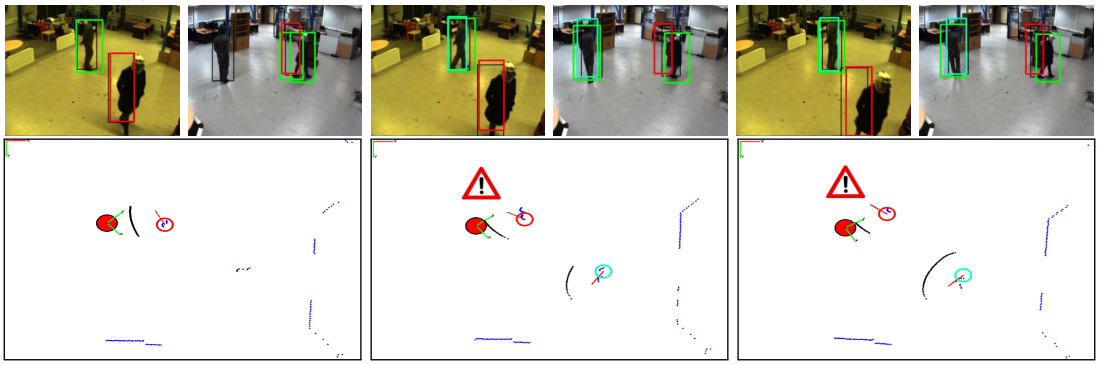
In conclusion, this chapter presented implementation details of a cooperative perception system between wall mounted fixed-view cameras and sensors on a mobile robot. The perception system comprises of detection and multi-person tracking. The cooperation is realized by fusing percepts from the deported and on-board sensors in a centralized manner via the RJMCMC-PF. The thorough off-line evaluations carried attest the improvements brought by the proposed cooperative

fusion. The framework has also been used to demonstrate the perception improvements brought by the BIP based HOG detector (HOG-BIPBoost) presented in the first part of the thesis owing to its improved frame rate. Finally, it has been demonstrated, again via on-line evaluation, the perceptions can be used endow the mobile robot safe navigation capabilities, without violated people's personal spaces, in people occupied environments.

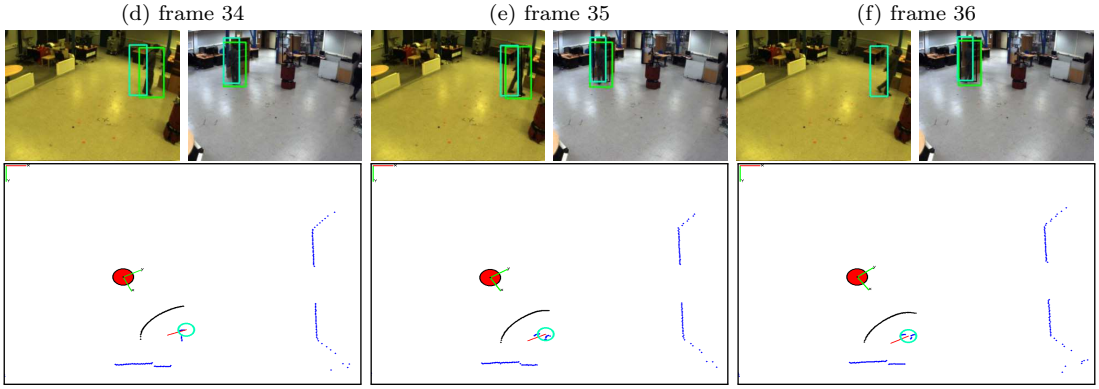
In its current form, this system could be extended for an area covered by similar number of cameras without problem. But, the system will have a problem if there be need to increase the number of cameras greatly for we have adopted a centralized fusion approach. At each cycle, information has to be sent to the central processor which will create a bottleneck when done over a network connection for large number of cameras. In summary, the presented system is well adapted for moderate size areas that necessitate the increased perceptual accuracy. Portions of the work presented in this chapter have been published in the following papers: [Mekonnen 2013c, Mekonnen 2013b, Mekonnen 2013d, Mekonnen 2012].



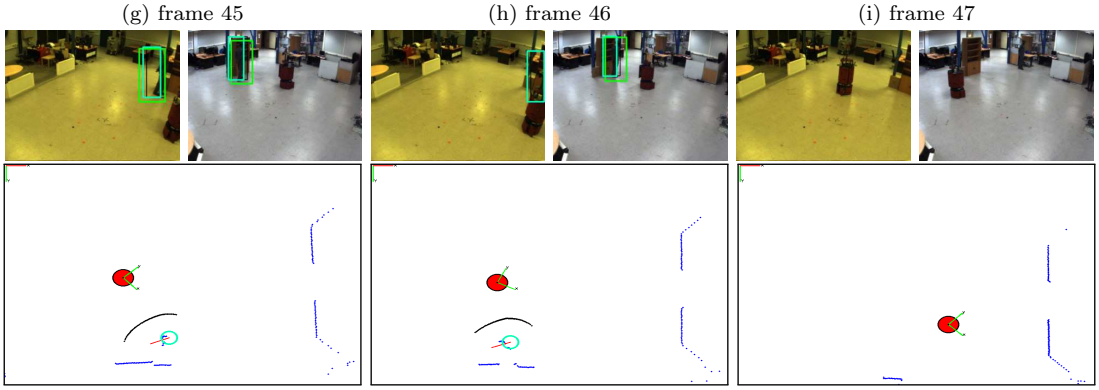
(a) frame 26 (b) frame 28 (c) frame 33



(d) frame 34 (e) frame 35 (f) frame 36

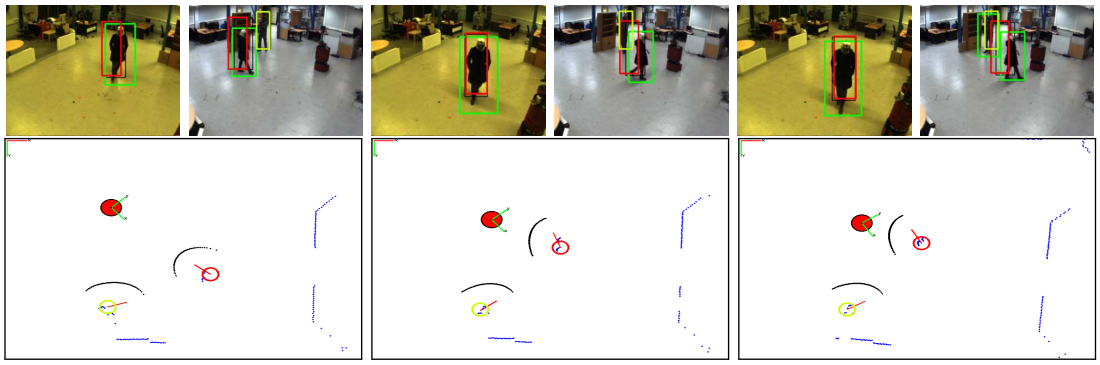


(g) frame 45 (h) frame 46 (i) frame 47

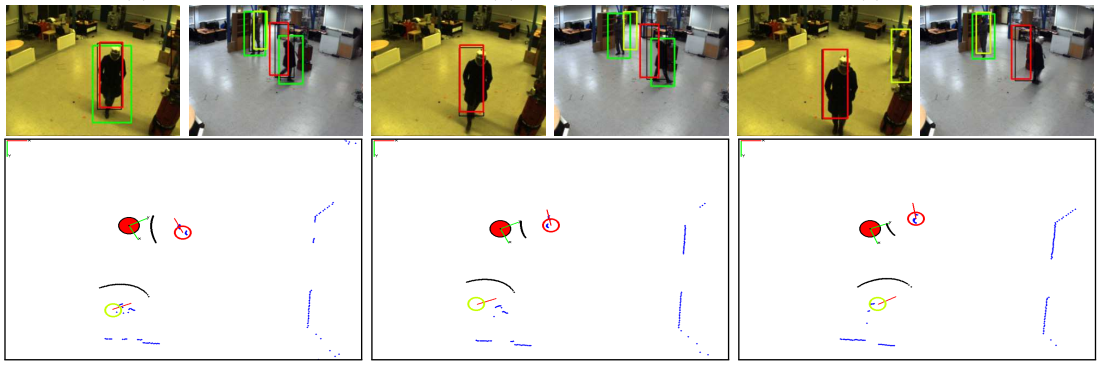


(j) frame 48 (k) frame 53 (l) frame 88

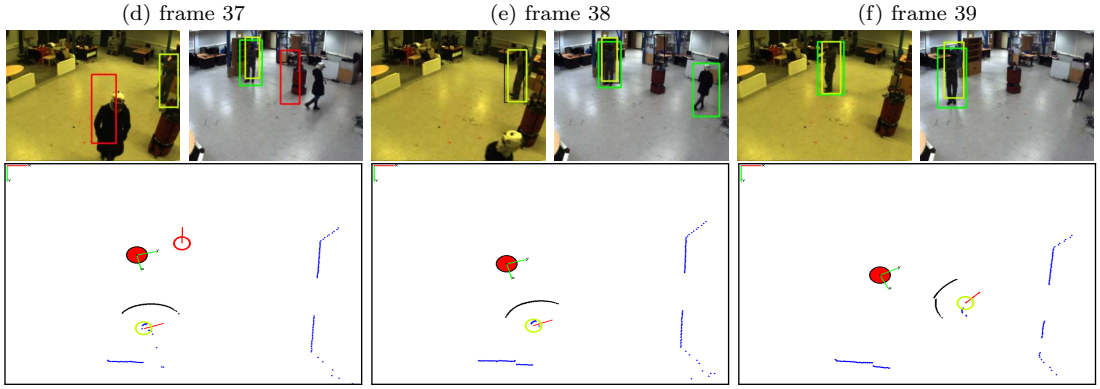
Figure 6.15: Sample robotic run for safe navigation evaluation. Perception is based on the GPU-HOG detector. (Please see text for explanation).



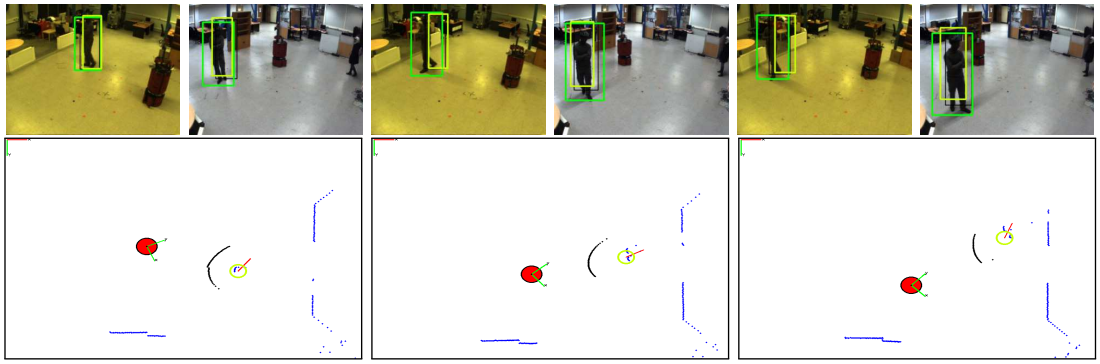
(a) frame 31 (b) frame 35 (c) frame 36



(d) frame 37 (e) frame 38 (f) frame 39



(g) frame 40 (h) frame 44 (i) frame 56



(j) frame 58 (k) frame 63 (l) frame 66

Figure 6.16: Sample robotic run for safe navigation evaluation with perception based on the GPU-HOG-BIPBoost detector. (Please see text for explanation).

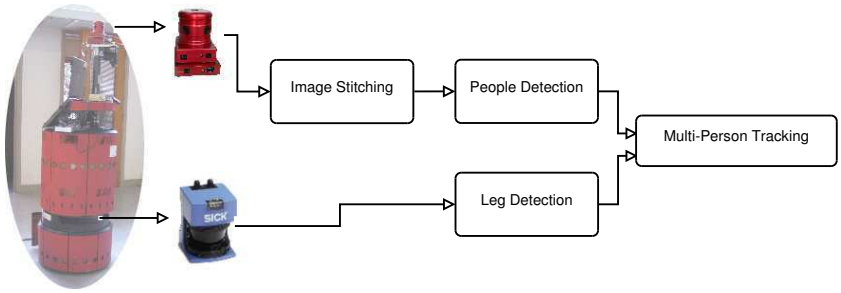


Figure 6.17: A robotic self-contained perceptual system using a spherical camera, *the Ladybug2*, and laser range finder.

Conclusions and Future Prospects

Globally, this thesis focused on the development of automated people perception modalities, namely multi-person detection and tracking, in the context of automated public place surveillance. In it we strongly contend the inclusion of a mobile robot in public place surveillance systems, for it can be used as a mobile sensor unit and a means for action—very useful amenities to pave the way towards a fully automated surveillance system. We start off our investigations by designing an automated visual people detector that is optimized with respect to detection performance and computation time explicitly to account for detection requirements and computational constraints faced by robotic and multi-sensor surveillance systems. We then use this novel detector and other percepts to realize a cooperative multi-person tracking system between wall mounted cameras and a mobile robot. We also show how the mobile robot can benefit from these perceptions by using the outcome to realize a navigation scheme that takes the safety of the people around into consideration. Additionally, we highlight an on-going development based on sensors solely on-boarded on a mobile robot. The two modes, the cooperation between wall mounted and mobile sensors and the self sustained autonomous mode, pave the way towards a generic surveillance system that addresses the pressing issues of automated complex environment monitoring in part. At all times, the proposed modes have been backed up with detailed evaluations that showcases the improvements brought by each one.

This manuscript summarizes the different investigations carried out in six chapters divided into two main parts. In the first chapter, it provides an overview of the problematic and details the objectives of the work and relevant contributions made by it. The first part of the thesis then presents the work on automated visual people detection whereas the second part focuses on cooperative multi-person tracking.

In the first part of the manuscript, chapters two to four deal with automated people detection aspects of the work. In the second chapter, we give a comprehensive review of the different trends, modes, and considerations in automated people detection. It is intended to put the materials presented in chapters three and four in perspective with respect to the state-of-the-art. In the third chapter, we presented our first people detector based on the proven discriminant HOG features in a cascade configuration. We propose and thoroughly validate a novel feature selection technique based on Binary Integer Programming (BIP) to select features that achieve a stipulated detection performance with the least cumulative computation time. The final detector achieved a significant boost in frame rate with comparable detection performance to Dalal

and Triggs [Dalal 2005] HOG detector. We achieve further improved frame rate of the learned detector via a GPU implementation.

In the forth chapter, we increased the feature pool to consider heterogeneous pool of features comprising Haar like features, EOH, HOG, CSS, and CS-LBP. Consequent to the significant variation in computation time and complementarity between the features, we investigated several approaches—Pareto-Front analysis, random feature selection, and BIP in conjunction with classical adaboost—and—a computation time weighted adaboost variant—for feature selection and classifier learning. We carried out several test experiments to investigate the advantages and shortcomings of each strategy using proprietary and multiple public datasets. The obtained results ascertain that complementary heterogeneous features lead to improved detection performance, and under explicit consideration of computation time, lead to improved frame rate as well. The results also attest the superiority of the BIP based feature selection strategy. The final detector based on BIP achieves an estimated frame rate that is amongst the best in the state-of-the-art, and logically outperforms our HOG-BIPBoost version. The BIP based framework also has an appealing characteristics in its flexibility with respect to computational resource constraints and detection requirement. On any dataset, the stipulated detection parameters used during training can either be made stringent or relaxed to learn a model that can, accordingly, consume more computational resource or less.

The second part of the thesis focuses on cooperative strategies for multi-person tracking. In this vein, it starts out in chapter five with an overview on cooperative people detection and tracking works in the literature and an overview on multi-person tracking. It then presents the formulation of the RJMCMC-PF multi-person tracking with a concise justification of why we have chosen to use it for the tracking application. In chapter six, we primarily present a new framework based on centralized cooperative strategy between wall mounted fixed view cameras and a mobile robot for tracking multiple persons in a surveilled area. This is a centralized cooperation strategy that fuses the image data from the wall cameras and laser range finder data from the mobile robot using an RJMCMC-PF to track individuals in the scene in time. We carry out several off-line and on-line experiments and attest the following. First, superior tracking results are obtained when fusing the inputs from the deported and mobile sensors, which asserts the advantages of using this cooperative system for surveillance of moderate-area that require increased accuracy, like high security zones. Second, the frame rate improvement brought by our proposed detector in Part I, does indeed improve the tracking accuracy and precision, specially with fast moving targets. Third, we show how the mobile robot can benefit from the cooperative perception during navigation to realize people safe navigation by respecting the personal space of each individual in the scene. We also show our proposed people detector consequently improves the mobile robot's reactivity, as a consequence of the perception frame rate improvement, leading to lessened people's personal space violations by the robot during navigation. In this chapter, we also presented a brief description of an ongoing development to realize a similar perceptual system using an omnidirectional camera (the *Ladybug2*) and a laser range finder mounted on our robot. This is envisaged to realize a stand alone mobile perception capabilities for covering dead-spots and other areas not covered by the FOV of conventional wall mounted cameras.

All in all, the different investigation and/or development carried out during this Ph.D. thesis resulted in the publications of one international journal, five international conference papers, one national journal, and one national conference. Additionally, it also resulted in one international journal publication in collaboration. The future prospects pertaining to it can be considered as short term, mid term, and long term prospects. At short term, various prospects are considered. The first, which is actually on-going at the moment, relates to the development to realize similar perceptual system using sensors on-boarded on a mobile robot. The actual development concerns a *Ladybug2* visual camera system that has 360^o FOV and a laser range finder mounted on

our robot. This system will be capable of providing surveillance activities, acting as a mobile sensor unit, on areas not covered by environment fixed cameras. This will complement the cooperative strategy presented in chapter six increasing the possible operation modes to cover complex environments. The second prospect is to implement a GPU version of the HETERO-BIPBoost detector presented in chapter four to further boost the people's detector frame rate. Current frame rate estimates of this detector has been presented in chapter six, but the prospect would actually be to finalize the GPU version and see how it affects the overall perception capabilities in the vein of the on-line experiments presented in chapter six. The third prospect concerns investigation with the control law aspects related to robotic navigation and passers-by (people) avoidance. In this work, the focus has been on perception aspects and not on control aspects. As a result, we have used the Nearness Diagram (ND) navigation off the shelf with some input modifications to accommodate tracked targets' security zone. But, we feel time should be invested to investigate control laws based on visual servoing techniques, *e.g.*, [Cadenat 2012], to realize a more application specific advanced controller for: (1) passers-by avoidance task, (2) person following (part of the service provision), *etc*.

Mid term prospects envisaged are concerned with further investigation on the BIP framework presented in Part I of this thesis. The investigations include: (1) investigation with other boosting variants, *e.g.*, real AdaBoost, Gentle Boost, *etc*.; (2) adding additional complementary features in the feature pool, *e.g.*, co-occurrence features (this actually can be considered as a short term prospect); (3) incorporate additional feature characteristics in the optimization, *e.g.*, feature stability as discussed in [Zhu 2006]; and (4) training a detector for a user specified frame rate, the BIP formulation can be modified slightly to optimize over the complete cascade to result in a detector that strictly meets a specified frame rate constraint with the highest possible (under that frame rate) detection performance.

The future, long term, prospect would be: to investigate event/behavior recognition based on the multi-person tracking results, and to investigate at the supervision level to move towards a fully automated surveillance system.

Appendix A

Brief Description of Sensors used for People Detection

In the literature, a variety of sensors have been employed for automated people detection. All these sensors can be boldly categorized into active and passive sensors. Active sensors work by radiating some sort of radiation on to the object/scene and provide measurement information inferred from the reflected radiation. On the other hand, passive sensors provide measurement information that is directly obtained from the levels of energy that are naturally emitted, reflected, or transmitted by the object/scene. Putting budgetary issues aside, the specific choice to use a distinct type of sensor is motivated by the application context: required information and environment interference. In the following subsections, a variety of active and passive sensors used for automated people detection are briefly presented along with their pros and cons.

A.1 Passive Sensors

Passive sensors measure level of energy that are naturally emitted, reflected, or transmitted by objects/scene. The common passive sensors used to detect people are visible spectrum cameras, thermal cameras, and microphones.

Visible Spectrum Cameras These cameras capture light in the visible electromagnetic spectrum (0.4 - 0.74 μm) by making use of matricial imaging chips, either CCD (Charge Coupled Device) or CMOS (Complementary Metal Oxide Semiconductor) chips. These sensors capture rich spatial information, horizontally and vertically; they could even be made to capture a scene with more than 50% of a complete spherical field of view with the help of special lenses and extra configuration [Yasushi 1999]. They are cheap, versatile, and provide color and texture information of the scene which is very useful during analysis. Consequently, they are the most used sensors for people detection with overwhelming methodologies [Dollár 2012, Gerónimo 2010a]. Their main shortcoming, inherent to their basic working priniciple, is their dependence on lighting conditions. They are hardly useful in low lighting, foggy, and invisible environments. Visible

spectrum cameras could be configured in pair to make a stereo camera system which allows 3D perception of the scene via triangulation. This added 3D information is very useful for possible candidate detection generation by 3D blob segmentation, though, further verification using 2D image is required for acceptable performance [Muñoz-Salinas 2007, Gerónimo 2010b].

Thermal Cameras These are cameras that capture electromagnet radiation with a wavelength in the range 6-15μm. Basically, they capture electromagnetic radiation emitted by objects in the scene which is a function of the objects' body temperature. Due to their body temperature, people leave peculiar bright profile on thermal images. As a result, people detection can be achieved via image segmentation, noise filtering, and a series of morphological operations [Correa 2012, Treptow 2005]. These cameras are quite useful for people detection in environments with low visibility. But, they are severly affected by elevated ambient temperature or other warm objects in the scene. As a result they are suitable for indoor usages, with controlled environment, and unsuitable for outdoor usage, where there are abundant warm bodies, *e.g.*, vehicles, and uncontrollable ambient temperature.

Microphones Microphones are another form of passive sensors that have been used for people detection. These sensors change mechanical vibration caused by sound into electrical signal. By using a set of microphones placed apart, a minimum of two in number, it is possible to detect sound source direction by analyzing the signal arrival time difference between the different microphones [Huang 2008b]. Using this technique, coarse location of a speaking person can be inferred. But, this sensor can not be used alone for conclusive people detection for, (1) it can not detect silent people, and (2) it can not differentiate between human and non-human sounds unless voice recognition techniques are utilized. Nevertheless, it has been successfully used for people detection under these constraints especially in robotic contexts, *e.g.*, [Brückmann 2006, Bennewitz 2005].

A.2 Active Sensors

Active sensors transmit a signal and observe the reflection from the target. Most of the sensors in this category measure the time it takes for the reflected signal to reach the source and infer presence and/or distance of an object using this delay. An exception worth mentioning here is the newly burgeoning Kinect [Microsoft 2010] sensor which uses the principle of structured light for 3D perception. The rest of the sensors can be differentiated by the type of wave they use for scene illumination. The most common ones are: lidars (**li**ght **d**etection **a**nd **r**anging), which use light waves; radars (**ra**dio **d**etection **a**nd **r**anging), which use radio waves; and sonars (**so**und **n**avigation **a**nd **r**anging) which use sound waves.

Structure Light based RGB+D These sensor work using the principle of structured light for 3D perception [Salvi 2010]; it projects a known structured pattern onto a scene and infers the depth from the deformation of the pattern captured by the a camera. A popular example in this category is Kinect [Microsoft 2010]. Kinect is a sensor that is composed of an IR projector, an IR camera, and an RGB camera. The projected structured pattern is in IR and the deformation is captured by an IR camera. The RGB camera is used to provide additional color information of the scene. This kind of cameras are often called RGB+D cameras (RGB for color and D for depth). Due to the 3D information, trivial techniques involving blob segmentation with background subtraction and morphological operations can be used [Salas 2011]. Alternatively, some authors, *e.g.*, Spinello *et al.* [Spinello 2011], have shown people detection using 3D features

extracted from the 3D data and a statistical classifier trained to detect people based on these features. Figure A.1 shows the Kinect sensor composition and sample people detections. Compared to similar matricial 3D sensors like stereo pair, swiss ranger, *etc*, the Kinect is superior with respect to depth precision and cost. As a result it is extensively being researched, especially in indoor robotic applications. Its limitations arise from its limited nominal working depth range, $0.8m$ to $3.5m$, which discards too close and far objects, and interference from ambient IR waves in outdoor scenes.

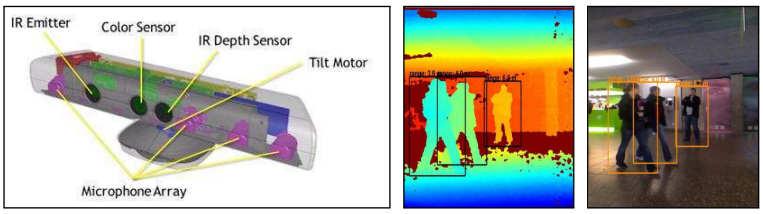
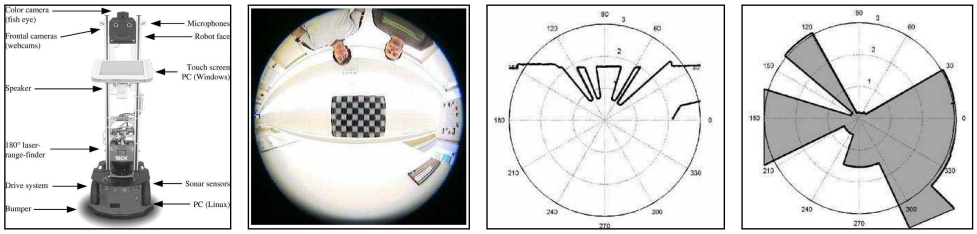


Figure A.1: Kinect: sensor (left), depth image (middle), and RGB image with detections (right). (Sensor image from [MSDN], and depth and RGB images from [Spinello 2011].)

Lidars Lidars emit electromagnetic waves, specifically infrared waves, and measure the time it takes for the reflected waves to arrive to determined distance of an object. These sensors are acclaimed for their distance measurement precision, speed, and perception under low visibility conditions. As a result, differing variants of lidars are extensively used in robotics. For people detection, a 2D variant that uses a single laser beam with a rotational mechanism, commonly called a 2D laser range finder (LRF), has been widely used. In few occasions a 3D variant known as a flash lidar camera–a camera that captures range and intensity in a pixel array of photodiodes–has also been used. With these flash lidar cameras detection techniques used on any other 3D sensor could potentially be used, for instance, Bennewitz *et al.* [Ikemura 2011] showed detection using distance based blob segmentation technique. Flash lidars are scarcely used for people detection because of their narrow field of view, exacerbated noise at far distances, and steep cost.

2D laser range finders provide depth data on evenly spaced angular positions within their 2D planar field of view. For a scanned person, the scan will exhibit a geometric shape corresponding to the person at the scanned height. Hence, detection can be performed by segmenting these shapes. In the literature, geometric cues like internal angles, curvature, diameter,*etc*, have been used to detect peoples' leg either using a set of observed bounds [Xavier 2005] or a statistically trained classifier based on these geometric features [Arras 2007]. In non cluttered and very well structured rooms, pairs of local minima corresponding to legs (figure A.2c) could be used as a detection cue [Martin 2006]. Multiple laser range finders can also be used at different heights to determine scans at different heights which combined would lead to robust detection. A good example is the work of Carballo *et al.* [Carballo 2009] in which chest and leg scans are used. The main downside of 2D lasers scanners is the lack of rich discriminating information amongst multiple people and the fact that they are easily fooled by similar structures in the scene. As a result they are mostly used combined with a visual camera, *e.g.*, [Mekonnen 2011, Kobilarov 2006, Cui 2005]. Compared to radars and sonars, they are preferable due to their speed, precision, and immunity to cross-talk.

Sonars Sonar sensors use the delay in the arrival of a reflected sound wave, initially emitted from the sensor, from an object to determine its range and direction. To cover wide field of



(a) (b) (c) (d)

Figure A.2: Illustration of sensor data from multiple sensors mounted on a mobile robot. (a) Mobile robot with mounted sensors, (b) image captured by the fish eye camera, (c) 180^o laser scan showing local minima corresponding to the two persons, and (d) sonar reading. (Images taken from [Martin 2006].)

view, a series of sonar sensors oriented in different directions will have to be used. Similar to laser range finder, people detection is achieved by filtering the data using geometric constraints (looking for narrow circular patterns for legs, figure A.2d) [Martin 2006]. Unfortunately, these sensors are not a popular choice due to their measurement inaccuracies; the are affected by an object's distance, direction, and surface absorption coefficient (which affects reflection); they are also prone to crosstalk effects in the presence of other sonars.

Radars Similarly, radars use the time of arrival of emitted then reflected radio waves to determine the distance to objects in the scene. People are detected by analyzing the distance profile in the field of view. In the literature, segmentation via clustering of reflected points [Milch 2001] and segmentation by analyzing peoples' radio wave reflection characteristics, average value of the reflected radio wave intensity [Yamada 2005], have been used for detection. Another technique, used to detect moving people from stationary radar sensors, is to apply background subtraction to segment out moving foreground people [Zetik 2006]. These sensors have demonstrated robust performance in virtually all weather conditions making them appealing for outdoor use [Zetik 2006], especially in the presence of fog or heavy rain. But, similar to laser range finders and sonars, they lack discriminating information amongst different people and provide misleading/confusing background scans in cluttered environment.

RFID reader Radio Frequency Identification (RFID) is a technique in which an RFID reader uses radio electromagnetic waves to detect the presence and identification of an RFID tag. In robotics, this has, for example, been used for environment mapping and localization by placing RFID tags all over a room [Hahnel 2004]. By making people wear unique RFID tags, the position of the tag (hence, their position), and unique id can be detected. An interesting work in this vein is the work of Germa *et al.* [Germa 2010]. In their work, a robot is equipped with 8 RFID readers oriented at distinct angular positions to span 360^o and multiplexed to detect a tag in the robot's vicinity. Using RFID technology for people detection is quite appealing but it can only be applied on people wearing the tag, which requires prior arrangements. In addition, it is not very suitable in crowded environments due to radio wave absorption unless combined with another sensor [Germa 2010].

Appendix B

Calibration Values: External cameras and a mobile robot

The following are the caliberation parameters (values) determined for the cooperative system described in chapter 6. The notations are determined using figure 6.3 as a reference.

$$A_1 = \begin{bmatrix} 611.7 & 0 & 311.7 \\ 0 & 611.8 & 269.7 \\ 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 605.5 & 0 & 358.8 \\ 0 & 605.9 & 260.0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.1}$$

$$^{c_1}T_w = \begin{bmatrix} -0.830 & -0.552 & 0.084 & 2.405 \\ -0.371 & 0.432 & -0.822 & 3.433 \\ 0.417 & -0.714 & -0.563 & 0.698 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{B.2}$$

$$^{c_2}T_w = \begin{bmatrix} -0.553 & 0.828 & 0.095 & 4.350 \\ 0.316 & 0.314 & -0.895 & -0.408 \\ -0.771 & -0.465 & -0.435 & 7.655 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{B.3}$$

Appendix C

Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU (publié dans RFIA 2014)

Résumé

Cet article présente un nouveau détecteur de personnes utilisant une sélection de descripteurs par optimisation discrète type branch and bound. Plus précisément, nous utilisons une programmation binaire pour sélectionner un sous-ensemble de descripteurs hétérogènes qui optimisent conjointement les performances en détection et le coût CPU. La mise en oeuvre de ce détecteur puis son évaluation sur des bases publiques montre clairement que cette reformalisation offre un bon compromis entre taux de faux négatifs et temps de calcul comparativement aux détecteurs existants de la littérature.

Mots Clef

Détection de personnes, sélection de descripteurs, apprentissage.

C.1 Introduction

De nombreuses applications s'appuient aujourd'hui sur des techniques avancées de vision par ordinateur. La détection visuelle de personnes *i.e.*, via une caméra perspective est certainement la plus usitée car ce capteur optique est bas coût, non intrusif, et délivre une information très

riche sur la scène observée (couleur, texture). Citons ici les applications de vidéosurveillance, interaction homme-machine, robotique, automobile, indexation d'images, etc. Un enjeu est le coût CPU et la robustesse du détecteur à divers artefacts : variations d'apparence des personnes, du point de vue, d'illumination, voir mouvement du capteur si celui-ci est embarqué. Certes, des avancées notables [Dollár 2012] ont été observées dans la communauté Vision mais cet enjeu reste encore aujourd'hui d'actualité.

Notre approche vise ici à prendre en considération explicitement le coût CPU dans le processus de sélection des descripteurs sous-jacents au détecteur. Ce coût est vital dans tout système réel *e.g.*, en robotique où la réactivité du système est conditionnée par les ressources CPU embarquées et les temps de traitement. Ces temps de traitement peuvent être prohibitifs notamment pour des capteurs optiques de dernière génération, *e.g.*, la caméra Ladybug de Point Grey [Point Grey Inc. 2012] qui exhibent des résolutions pixel élevées. Il est vital alors de privilégier pour le détecteur des descripteurs discriminants mais aussi peu coûteux en CPU. Ce compromis est en pratique difficile à obtenir. Ainsi, les histogrammes de gradients orientés (HOG) [Dalal 2005] sont des descripteurs très discriminants mais très coûteux comparativement aux descripteurs de type Haar [Viola 2004]. Certes, les dernières avancées considèrent des détecteurs mixant des descripteurs hétérogènes (HOG, Haar, etc.) [Walk 2010, Wojek 2008] ou modélisant explicitement/implicitement [Felzenszwalb 2010b] les parties corporelles... mais toujours au détrimant du coût CPU qui n'est pas explicité dans la formulation. Ce constat a motivé nos travaux qui visent à développer un détecteur offrant un compromis entre taux de classification et coût CPU.

Travaux antérieurs : La littérature propose de nombreux détecteurs de personne et un état de l'art détaillé serait ici superflu; le lecteur pourra se référer ici à [Dollár 2012]. Notre étude se limitera ici aux investigations privilégiant un ensemble hétérogène de descripteurs et une technique de fenêtre glissante pour générer les échantillons/régions à classer. Cette démarche, en mixant des informations complémentaires, améliore les performances à l'instar de Dollar *et al.* dans [Dollár 2012].

Citons également Wojek *et al.* [Wojek 2008] qui mixent des descripteurs de type Haar, HOG, et *shape context*. Leur étude comparative à partir de classifieurs SVMs ou *boosting* montre clairement que la fusion de descripteurs hétérogènes est plus performante et donc supplante les approches se bornant à un pool de descripteurs homogènes. Walk *et al.* dans [Walk 2010] ont abouti au même constat en concaténant HOG, histogramme de flot optique [Dalal 2006b], et *Color Self Similarity* (CSS).

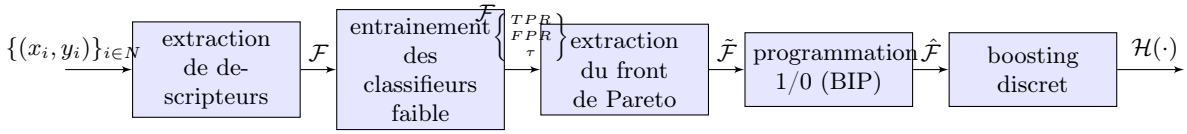


Figure C.1: Schéma du synoptique de l'apprentissage du classifieur fort propre à chaque nœud de la cascade.

Quatre stratégies de fusion des descripteurs hétérogènes sont alors privilégiées dans la littérature pour construire le détecteur :

1. Une concaténation directe des descripteurs [Walk 2010, Wojek 2008] induisant un fort coût CPU de par la complexité du descripteur final et les poids du classifieur associés dans la détection par fenêtre glissante.

2. Un *boosting* direct [Wojek 2008, Gerónimo 2007] *i.e.*, chaque classifieur fort apprend directement et itérativement le sous ensemble des descripteurs pertinents parmi le pool com-

plet hétérogène. Hélas, à chaque itération, un descripteur est sélectionné par le classifieur indépendamment de son coût CPU. Cette démarche privilégie les descripteurs certes discriminants mais complexes et donc augmentant le coût CPU.

3. Un arrangement hierarchique [Mogelmose 2012, Pan 2013] *i.e.*, la cascade multi-classifieurs considère des descripteurs à faible coût CPU dans ses étages initiaux et des descripteurs plus complexes dans ses étages supérieurs. Cette démarche offre un compromis entre taux de détection et vitesse. Certains travaux [Mogelmose 2012, Pan 2013] s'appuient ici sur des heuristiques et des familles homogènes de descripteurs simples et complexes respectivement pour les étages initiaux et suivants.

4. Un compromis entre vitesse et taux de détection à l'instar des travaux menés dans Wu and Nevatia [Wu 2008], Jourdheuil *et al.* [Jourdheuil 2012], et Mekonnen *et al.* [Mekonnen 2013f]. Le principe est de combiner au sein d'un même critère, et avec des pondérations dédiées, le coût CPU et les performances de détection. Cette formulation masque les contributions de chacun des deux critères sous-jacents et n'offre aucune garantie d'optimalité.

Notre approche s'inscrit dans cette dernière stratégie mais elle sélectionne à chaque nœud de la cascade les descripteurs optimisant conjointement et distinctement les deux critères pré-cités. Nous considérons quatre familles usuelles de descripteurs : Haar [Viola 2004], Histogramme orientation des contours (EOH) [Gerónimo 2007], CSS [Walk 2010], *Center Surround Local Binary Patterns* (CS-LBP) [Heikkilä 2009], et HOG [Dalal 2005] dans un cadre de *boosting* structuré en cascade [Viola 2004] avec une optimisation discrète basée sur une programmation 1/0 (BIP) sélectionnant le sous ensemble des descripteurs offrant le meilleur compromis coût CPU-performance.

Contributions : Cet article propose une reformulation du processus d'optimisation, ici BIP, et prenant en compte coût CPU et performance de détection dans le processus de sélection des descripteurs. Cette reformulation est clairement novatrice dans la littérature. Des évaluations sur la base d'images publiques INRIA [Dalal 2005] sont ensuite proposées afin de quantifier les gains obtenus comparativement aux détecteurs existants de la littérature.

C.2 Descriptif de notre approche

Pour rappel, l'objectif est de prototyper un détecteur basé sur des descripteurs capturant l'aspect visuel d'un individu et ceci indépendamment du point de vue de la caméra, de l'apparence, de l'illumination, etc. Bref, l'apprentissage hors ligne vise à sélectionner un sous-ensemble de descripteurs discriminants au mieux une silhouette humaine générique... et peu onéreux en CPU.

Nous privilégions, pour son faible coût CPU et à l'instar de [Viola 2004], un mécanisme de cascade attentionnelle classant en positifs (humain) ou négatifs (autres) des sous-images générées par une technique de fenêtre glissante dans l'image entière. L'apprentissage du classifieur fort propre à chaque nœud de la cascade est schématisé par le synoptique figure C.1. Soient n échantillons positifs ou négatifs d'apprentissage notés $\{(x_i, y_i)\}_{i\in\{1,\dots,n\}}$), les descripteurs listés en section § C.2.1 sont extraits, l'ensemble associé est noté $\mathcal{F}$. Pour chaque descripteur, un classifieur faible est entraîné à partir de la base d'apprentissage afin de caractériser son pouvoir discriminant en termes de taux de vrais positifs (TPR) et taux de faux positifs (FPR). Puis, une analyse par front de Pareto permet de sélectionner un sous-ensemble de descripteurs notés $\tilde{\mathcal{F}}$, et prenant en considération les critères TPR, FPR, et coût CPU. Cette étape est vitale pour réduire de façon drastique le nombre de descripteurs candidats... en préambule à l'étape d'optimisation discrète. Cette étape d'optimisation, détaillée en § C.3, est exécutée pour sélectionner un sous-ensemble restreint de descripteurs $\hat{\mathcal{F}}$ ayant le meilleur compromis performance-coût CPU. Au

final, un classifieur fort par nœud $\mathcal{H}(\cdot)$ est entrainé à partir de ce sous ensemble de descripteurs $\hat{\mathcal{F}}$ par une technique de *boosting* discret. Chaque bloc du synoptique est détaillé ci-après.

C.2.1 Les descripteurs

Au total, cinq familles de descripteurs sont considérées, qui sont : Haar, CS-LBP, CSS, EOH, et HOG. Ce choix est motiver par deux aspects: (1) leur usage fréquent dans la littérature pour la détection de personne, et (2) leur complémentarité. EOH et HOG capturer distributions de bord, CSS se concentre sur la couleur symétrie, Haar et CS-LBP sur l'intensité et les variations de texture. Les descripteurs entières de chaque famille sont extraites en utilisant un 128×64 pixels de la fenêtre de modèle humain.

Haar : Ici, l'ensemble étendu proposé par Lienhart et Maydt [Lienhart 2002] qui comprend les variantes inclinés est utilisé. L'ensemble complet est recueilli par extraction des valeurs de descripteurs à tous les postes et les échelles de la fenêtre de modèle.

CS-LBP : Calcule par pixel CS-LBP [Heikkilä 2009] valeur en prenant et en modulant la différence d'intensité de pixels centraux symétriques pour tous les pixels voisins. Pour chaque pixel, on privilège d'une région de pixels de 3×3 ce qui conduit à un nombre entier scalaire entre 0 et 16. Ensuite , un histogramme de bacs 16 est calculé compte tenu d'une zone rectangulaire. Cela signifie une descripteur de cette famille. Pour toutes les positions et les échelles possibles de la région rectangulaire, une descripteur distincte (qui est un histogramme) est calculé et ajouté à l'ensemble CS-LBP .

CSS : Le calcul commence d'abord par la subdivision de la fenêtre de modèle donné en blocs. Pour chaque bloc, une couleur histogramme HSV de 3x3x3 est construit. Ensuite, la similarité de bloc avec le reste des blocs est déterminé par l'intersection d'histogramme. Au lieu de la concaténation de toutes les similitudes calculés comme Walk *et al.* [Walk 2010], nous définissons un seul CSS descripteur comme un vecteur de valeurs scalaires qui sont obtenus par l'intersection de l'histogramme d'un bloc avec les autres blocs. L'ensemble de CSS desciprteur est alors déterminé par le calcul de ce vecteur pour tous les blocs. En divisant le modèle en blocs de 8×8 pixels, un total de 128 descripteurs, chacun avec 127 dimensions, sont obtenus.

EOH : Ce pool de descripteur est générée exactement comme décrit par Geronimo *et al.* [Gerónimo 2007] : histogramme de l'orientation du contour suivi par les ratios de magnitude de deux bacs pour obtenir une valeur scalaire unique et le faire pour toutes les positions et les échelles de sous-régions rectangulaires dans la fenêtre de modèle.

HOG : L'ensemble HOG est construit comme suit : Soit la fenêtre de modèle, il est divisé en blocs et un histogramme des gradients orientés de 36 dimensions est calculée comme [Dalal 2005]. Mais, plutôt que la concaténation de tous les histogrammes de blocs pour faire une descripteur de grande dimension, nous considérons la concaténation un sous-ensemble couvrant une zone rectangulaire. La famille de la descripteur HOG est généré en considérant toutes les positions possibles, la largeur et hauteur de la région rectangulaire. Les desctribeurs varient d'un vecteur de 36 dimensions, qui contient un seul bloc, à un vecteur de 3780 dimensions, qui contient tous les blocs dans le modèle.

Table C.1 présente les nombres total de descripteurs, maximale et minimale temps de calcul (τ_{max} et τ_{min}), et le classifieur faible utilisée dans chaque famille de descripteur. Pour le famille CS-LBP, analyse discriminante linéaire (LDA) associé à un arbre de décision (qui est construit après reprojection) est privilégié comme SVM mène à la période d'entraînement immense (en raison du nombre élevé de discripteurs CS-LBP).

Table C.1: Récapitulatif des descripteurs utilisés ; $\mathfrak{u} = 0.0535\mu s$.

descripteurs	nombre total	τ_{min}	τ_{max}	classifieurs faible
Haar	672,406	1.0$\mathfrak{u}$	3.48$\mathfrak{u}$	arbre de décision
EOH	712,960	4.83$\mathfrak{u}$	317.75$\mathfrak{u}$	arbre de décision
CS-LBP	59,520	15.45$\mathfrak{u}$	393.64$\mathfrak{u}$	LDA + arbre de décision
CSS	128	1017.94$\mathfrak{u}$	1017.94$\mathfrak{u}$	SVM
HOG	3,360	489.72$\mathfrak{u}$	51420.56$\mathfrak{u}$	SVM

C.2.2 Extraction du front de Pareto

Soit $\mathcal{F}$ l'ensemble initial de descripteurs, leurs classifieurs faibles associés avec trois paramètres sous-jacents : TPR, FPR, et coût CPU (noté τ). L'analyse par front de Pareto exhibe les solutions optimales au sens de ces paramètres. Le sous-ensemble associé de descripteurs constitue le front de Pareto optimal *i.e.*, on ne peut améliorer un paramètre sans dégrader un des deux autres : ce sous-ensemble de descripteurs, optimal au sens de Pareto pour les trois paramètres pré-cités, est noté $\tilde{\mathcal{F}}$; il est alors exploité par le processus d'optimisation discrète.

C.2.3 Sélection des descripteurs et apprentissage de la cascade

Le processus de sélection finale des descripteurs est piloté par optimisation discrète type BIP détaillé en § C.3. Cette étape génère le sous ensemble $\hat{\mathcal{F}}$ de descripteurs. Au final, le classifieur fort propre à chaque nœud $\mathcal{H}(\cdot)$ s'appuie sur ce sous-ensemble et une technique de Adaboost discrète.

Le classifieur complet est structuré autour de plusieurs nœuds formant la cascade. Sa construction s'appuie initialement sur tous les échantillons positifs et un sous-ensemble d'échantillons négatifs (en nombre équivalent aux positifs) pour apprendre les descripteurs relatifs au premier nœud/étage. Tous les négatifs sont alors testés sur ce premier nœud, les vrais négatifs sont rejetés tandis que les faus positifs sont conservés pour les nœuds suivants. La démarche est re-itérée jusqu'à traitement de tous les négatifs. Cette technique dite de *data mining* permet l'exploitation d'un nombre flexible de négatifs.

C.3 Optimisation discrète

Une sélection des descripteurs basée sur un programme linéaire en variables binaires (une programmation 1/0) constitue une contribution essentielle de ce travail. La formulation proposée vise à minimiser le temps de traitement dans la cascade de détection. Elle prend en paramètre les taux de vrais et faux positifs souhaités (TPR_k, FPR_k), à chaque étage k.

Définition des paramètres : La liste suivante indique les paramètres utilisés dans la formulation. $\mathbb{B} = \{0, 1\}$ est l'ensemble binaire.

- $N = \{1, ..., n\}$ est l'ensemble des échantillons avec $n \in \mathbb{Z}$; un échantillon étant référencé par l'index i;
- $M = \{1, ..., m\}$ est l'ensemble des classifieurs faibles avec $m \in \mathbb{Z}$; un classifieur faible étant référencé par l'index j;
- les vecteurs $\mathbf{y}^+ \in \mathbb{B}^n$, $\mathbf{y}^+ = \left\{y_i^+\right\}_{i \in N}$ et $\mathbf{y}^- \in \mathbb{B}^n$, $\mathbf{y}^- = \left\{y_i^-\right\}_{i \in N}$ indiquent la nature des échantillons :

$$y_i^+ = \begin{cases} 1 & \text{si } i \text{ est positif} \\ 0 & \text{sinon} \end{cases} \qquad y_i^- = \begin{cases} 1 & \text{si } i \text{ est négatif} \\ 0 & \text{sinon} \end{cases}$$

- $\mathbf{H} \in \mathbb{B}^{nxm}$ où $\mathbf{H} = \{h_{i,j}\}_{\substack{i \in N \\ j \in M}}$ avec $h_{i,j} \in \{0,1\}$ est la matrice de couverture des échantillons par les classifieurs faibles.

$$h_{i,j} = \begin{cases} 1 & \text{si le classifieur faible } j \text{ détecte l'échantillon } i \\ & \text{comme positif} \\ 0 & \text{sinon} \end{cases}$$

- $\mathrm{TPR}_k \in [0,1]$ est le taux minum de vrais positifs souhaité à l'étage (k) de la cascade;

- $\mathrm{FPR}_k \in [0,1]$ est le taux maximum de faux positifs attendu à l'étage (k) de la cascade;

- $\tau \in \mathbb{R}^m$, avec $\tau = \{\tau_j\}_{j \in M}$, désigne le temps de calcul associé au détecteur j.

Variables de décision : Les variables de décision sont binaires.

- $\mathbf{v} \in \mathbb{B}^m$, avec $v_j \in \{0,1\}$, définit l'ensemble des classifieurs faibles sélectionnés à l'étage k : $v_j = 1$ si le détecteur j est sélectionné, 0 sinon;

- $\mathbf{T} \in \mathbb{B}^n$, avec $t_i \in \{0,1\}$, correspond à l'ensemble des vrais positifs détectés : $t_i = 1$ si l'échantillon positif i est détecté positif par au moins un détecteur , 0 sinon;

- $\mathbf{F} \in \mathbb{B}^n$, avec $f_i \in \{0,1\}$, correspond à l'ensemble des faux positifs détectés : $f_i = 1$ si l'échantillon négatif i est détecté positif par au moins un détecteur , $f_i = 0$ sinon.

Nous introduisons le vecteur $\mathbf{p}$, $\mathbf{p} = \{p_i\}_{i \in N} = \mathbf{Hv}$ qui indique, pour chaque échantillon i, le taux total de détecteurs ayant détecté l'échantillon positif.

Formulation :

$$\begin{array}{lll} \min & \tau^{\mathrm{T}}\mathbf{v} & (1) \\ \text{s.t} & t_i \leq y_i^+ \cdot p_i \qquad \forall i & (2) \\ & f_i \geq y_i^- \cdot h_{i,j} \cdot v_j \qquad \forall (i,j) & (3) \\ & \|\mathbf{T}\|_1 \geq \|\mathbf{y}^+\|_1 \cdot \mathrm{TPR}_k & (4) \\ & \|\mathbf{F}\|_1 \leq \|\mathbf{y}^-\|_1 \cdot \mathrm{FPR}_k & (5) \\ & \mathbf{v} \in \mathbb{B}^n; \mathrm{T} = \{t_i\}_{i \in N}, \mathrm{F} = \{f_i\}_{i \in N}; \mathbf{T}, \mathbf{F} \in \mathbb{B}^n & (6) \\ & \|\cdot\|_1 \text{ est la norme } l_1. & \end{array}$$

La fonction objectif (1) a pour but de minimiser le temps de calcul total à l'étage k considéré. L'ensemble des contraintes (2)-(5) imposent qu'un certain niveau de qualité soit atteint (déterminé par les taux de vrais et faux positifs désirés). Les contraintes (2) font le lien entre les variables v_j et t_i (via p_i) : ainsi $t_i = 0$ si aucun détecteur sélectionné n'a identifié correctement l'échantillon positif i. Les contraintes (3) relient les variables v_j et f_i tel que $f_i = 1$ si l' échantillon négatif i a été reconnu positif par au moins un des classifieurs faibles sélectionnés. La contrainte (4) exprime qu'un taux de reconnaissance de TPR_k échantillons positifs doit être atteint. De façon symétrique, la contrainte (5) impose que le taux total de faux positifs ne doit pas excéder FPR_k. Le nombre total de contraintes dans cette formulation est égal à $(n \cdot (m+1)+2)$, ce qui peut être élevé lorsque des nombres importants de détecteurs (n) et d'échantillons (m) sont considérés. Nous appelons $\hat{\mathcal{F}}$ l'ensemble final des échantillons détectés positifs par les détecteurs sélectionnés.

C.4 Evaluations et résultats

Les évaluations menées dans ce travail sont axées sur les deux aspects suivants :
(1) L'évaluation de la stratégie de sélection de descripteur : Ici le but est d'analyser les avantages et inconvénients de l'utilisation de l'optimisation discrète de type BIP par rapport aux alternatives plus simples. La stratégie d'utiliser une sélection de descripteur basée sur l'optimisation BIP et un apprentissage par classifieur est comparée à deux autres modes. Le premier, appelé **Pareto+AdaBoost** supprime le bloc BIP du cadre du travail et entraine directement un classifieur fort à chaque nœud avec une technique d'Adaboost discrète utilisant les descripteurs retenues par le bloc d'extraction du front de Pareto. Le second, appelé **Random+AdaBoost**, construit directement un classifieur fort à chaque nœud en utilisant des descripteurs choisies aléatoirement depuis l'ensemble total des descripteurs (proportionnellement à la taille de chaque famille de descripteurs).
(2) Une évaluation générale par-rapport à l'etat de l'art : Dans cette partie, la performance de la méthode BIP+Adaboost est comparée aux méthodes principales de la littérature.

C.4.1 Critères d'évaluation

Pour évaluer la performance du détecteur, nous utilisons deux approches : (1) L'approche par fenêtre, où est générée une courbe DET (Detection Error Trade-off) représentant les faux négatifs par-rapport aux faux positifs par fenêtre (FPPW) en utilisant des fenêtres de taille réduite de positifs et de négatifs. La première courbe est utilisée pour comparer des variantes dde l'algorithme proposé par-rapport au détecteur HOG de Dalal et Triggs [Dalal 2005] *(aspect 1)*, et la seconde est utilisée pour déterminer comment se comnporte notre meilleure variante par rapport aux méthodes de la littérature *(aspect 2)*. Un taux de faux négatifs à 10^{-4} FPPW et une log-moyenne du taux de faux négatifs sont utilisés respectivement pour la première et la seconde approche.

Une autre critère à prendre en compte et le temps moyen de calcul. Pour un détecteur en cascade, le temps moyen de calcul pour une fenêtre candidate donnée dépend du FPR à chaque nœud. Soient K le nombre total de nœuds dans la cascade, FPR_k lle taux de faux positifs et τ_k le temps total de calcul du k^{ime} nœud pendant la détection. En supposant un taux de faux positifs d'une image d'entrée générique, le temps moyen passé sur une fenêtre-test candidate, $\mathcal{T}_{av}$, peut être estimé par $\mathcal{T}_{av} = \tau_0 + \sum_{k=1}^{K-1}(\prod_{z=0}^{k-1}\mathrm{FPR}_z)\tau_k$. En utilisant le détecteur de Dalal et Triggs [Dalal 2005] comme référence, qui prend un temps ζ_{HOG} par fenêtre, l'**accélération moyenne** (ASU) est donnée par $\mathrm{ASU} = \frac{\zeta_{HOG}}{\mathcal{T}_{av}}$. Par conséquent, les valeurs d'accélération moyennes reportées désormais sont calculées par-rapport au détecteur de Dalal et Triggs.

C.4.2 Jeux de données

Dans ce travail, en raison de contraintes de place, les résultats sont présentés sur un seu jeu de données public, la **base publique de données de l'INRIA** [Dalal 2005]. Il s'agit d'une base de données accessible au public utilisée principalement pour évaluer les performances des détecteurs de la littérature. Un total de 2416 fenêtres positives recadrées et de 2.55×10^6 fenêtres négatives uniformément réparties sont utilisées pour l'apprentissage. Pour l'évaluation par fenêtre, on utilise 1132 fenêtres positives recadrées et 2.00×10^6 fenêtres négatives uniformément réparties. Pour l'évaluation d'une image entière, la base de données fournit 288 images complètes annotées.

C.4.3 Apprentissage

Chaque nœud de la cascade d'apprentissage est régi par deux paramètres donnés : les TPR_k et FPR_k pour le nœud k (TPR_k vaut toujours 1.0). L'apprentissage est fait de telle sorte que le classifieur du nœud final soit conforme aux exigences de performance. Chaque nœud de la cascade est contruit en utilisant un sous-ensemble des échantillons négatifs d'apprentissage et tous les échantillons positifs. Cet ensemble est divisé initialement en deux sous-ensembles : 60% pour l'apprentissage et 40% pour la validation. Les classifieurs faibles sont entraînés en utilisant la base de données d'apprentissage. Ensuite, les valeurs de TPR et de FPR correspondand à chaque classifieur faible sont déterminées en se basant sur la base de données de validation. Tous les calculs suivants, c'est-à-dire l'analyse par front de Pareto et la sélection des descripteurs par BIP sont effectués en utilisant les performances des classifieurs faibles conférées sur la base de données de validation. Une fois que les caracteristiques pertinentes sont sélectionnées, les classifieurs faibles correspondant sont re-entraînés en utilisant à la fois la base de données d'apprentissage et celle de validation par une technique de boosting discret pour construire le classifieur fort final par nœud $\mathcal{H}(\cdot)$. Le classifieur complet en cascade est ensuite entraîné comme expliqué dans en § C.2.3. Pour les classifieurs faibles associés, des arbres de décision de profondeur 2, 3 et 3 sont utilisés respectivement pour les descripteurs de Haar, EOH après compromis entre les performances de détection et le sur-apprentissage sur l'ensemble de validation.

C.4.4 Résultats et discussions

Les résultats principaux obtenus avec la base de l'INRIA sont montrés sur la figure C.2 et sont présentés dans la table C.2. Nous avons entraîné deux variantes du classifieur BIP+AdaBoost. Dans le premier cas appelé **BIP+AdaBoost(Fix)**, un FPR par noeud de 0.5 est utilisé pour tous les noeuds. Dans un second cas, un FPR adaptatif est utilisé, en démarrant à 0.3 à l'étape initiale et en continuant les noeuds d'apprentissage, à chaque fois qu'une solution de l'optimisation par BIP n'existe pas, cette contrainte est relâchée en augmentant le FPR de 0.1 et la procédure continue jusqu'à ce que tous les échantillons négatifs soient épuisés. Cette version est appelée **BIP+AdaBoost(Ad)**. Les meilleurs résultats de détection à un FPPW de 10^{-4} sont obtenus par les variantes Random+AdaBoost et Pareto+AdaBoost. Les deux variantes de BIP+Adaboost surpassent le détecteur de Dalal et Triggs à 10^{-4} de plus de 2%. De plus, la méthode BIP+AdaBoost(Fix) atteint une accélération de la méthode de 15.6x tandis que BIP+AdaBoost(Ad) admet une accélération de 9.22x.

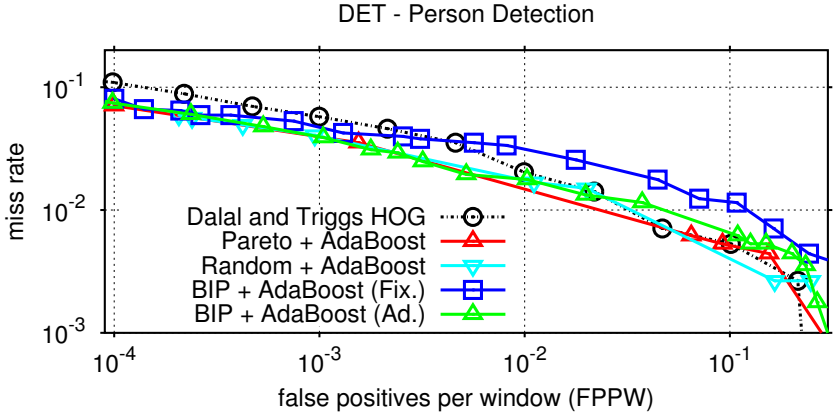
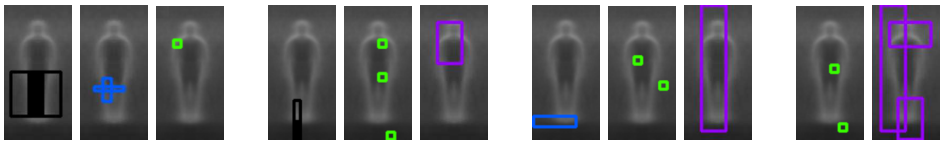


Figure C.2: DET des détecteurs entraînés et testés sur la base INRIA.



(a) nœud 0 (b) nœud 1 (c) nœud 2 (d) nœud 8

Figure C.3: Représentations d'exemples (en superposition sur une image moyenne de gradient humain) des descripteurs hétérogènes choisis dans les différents nœuds de la cascade formés en utilisant des données INRIA et FPR adaptative. Régions rectangulaires noires montrent descripteurs de Haar, bleu est pour CS-LBP, boîtes vertes représentent les descripteurs CSS et leur position indique le bloc de référence, et enfin, le violet montre la région de l'espace engendré par les blocs de HOG concaténés.

Comme les contraintes initiales de FPR sont strictes sur la variante BIP+AdaBoost(Ad), cela va favoriser les descripteurs discriminantes avec des temps de calcul augmentés. Mais cela va aussi contribuer à des performances de détection supérieures par rapport à BIP+AdaBoost(Fix), sur toute la gamme de FPPW présentée sur la figure C.2. Sur la table C.2, il y a une proportion plus importante de descripteurs de Haar (5.4% plus) et moins importantes de HOG (2.0% moins) dans la version fixe que dans la version adaptative, ce qui contribue à l'amélioration du temps de calcul.

Table C.2: Résumé du détecteur en cascade entrainé sur les bases de données de l'INRIA. Les taux de faux négatifs sont donnés à un FPPW de 10^{-4}.

détecteur	composition de descripteurs					MR	ASU
	Haar	CSLBP	CSS	EOH	HOG		
Dalal and Triggs [Dalal 2005]	–	–	–	–	100%	11.0%	1.0x
Pareto + AdaBoost	42.8%	14.5%	7.8%	25.6%	9.3%	7.0%	0.4x
Random + AdaBoost	26.3%	10.8%	3.7%	53.5%	5.6%	6.0%	0.4x
BIP + AdaBoost (Fix)	60.4%	10.8%	8.0%	9.7%	11.0%	8.0%	15.6x
BIP + AdaBoost (Ad)	55.0%	14.6%	8.1%	9.3%	13.0%	7.4%	9.22x

Sur la figure C.4 sont représentés les histogrammes des descripteurs sélectionnées, dans des proportions relatives, pour les premiers 9 noeuds des variantes fixe et adaptative de la méthode. Clairement, la variante fixe utilise des descripteurs moins coûteuses et augment le long de la cascade à la fois en nombre et en complexité. Au contraire, pour la variante variable, les descripteurs complexes apparaissent dans les noeuds initiaux et augmentent en nombre le long de la cascade. La figure C.3 illustre quelques descripteurs sélectionnées superposées à une image de gradient d'un humain pour la version BIP+AdaBoost(Ad). Nous pouvons remarquer que toutes les descripteurs sélectionnées représentent des facettes discriminantes de personnes.

Finalement, la figure C.5 présente l'évaluation comparative du détecteur BIP+AdaBoost(Ad) (la meilleure variante qui donne un bon compromis entre performance de détection et coût calcul) sur la base INRIA en utilisant les critères d'évaluation sur image complète. Les évaluations comparatives sont issues de [Dollár 2012] ; le lecteur pourra se referer à cete étude pour l'explication de chaque détecteur. Pour générer ces résultats, nous utilisons une suppression des non maxima par paire [Dollár 2012] avec un seuil de recouvrement de 0.65. Encore une fois, ici, la variante BIP+AdaBoost(Ad) réussit à une log-moyenne de faux négatifs de 47%. At des valeurs plus basses de FPPI, à moins de 0.1 FPPW, la variante BIP surpasse les HOG de Dalal and Triggs systématiquement. En utilisant les vitesses de calcul mentionnées dans [Dollár 2012] pour des personnes de plus de 100 pixels sur des images de taille 640×480, nos détecteurs arrivent à 2.3 images par seconde (fps) pour la variante adaptative, et à 3.9 fps pour la vairante à FPR fixe,

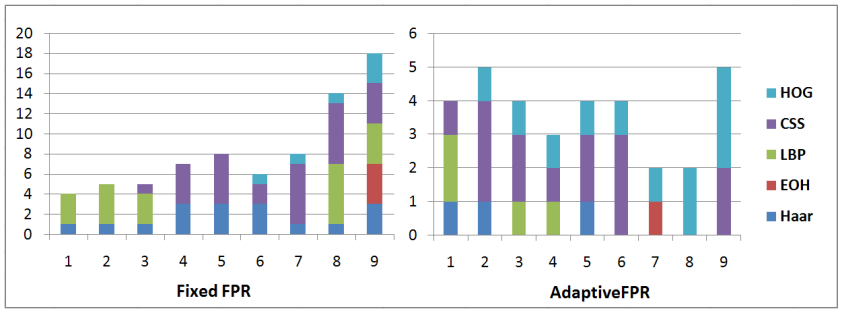


Figure C.4: Histogramme de descripteurs sélectionnées pour les 9 premiers noeuds des modèles entraînés sur la base INRIA avec un FPR fixe de 0.5 et avec un FPR adaptatif.

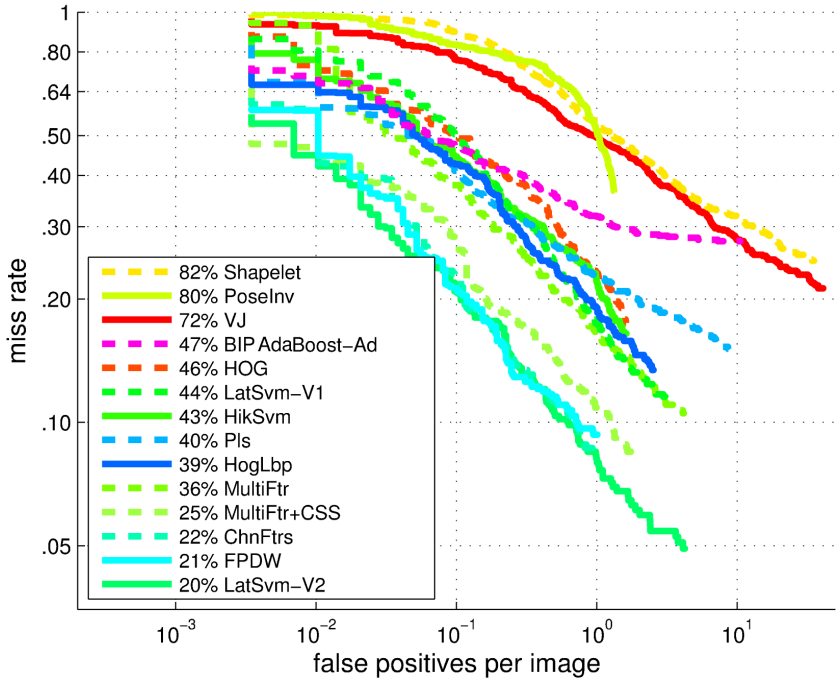


Figure C.5: Evaluation comparative avec images complètes sur la base test de l'INRIA.

entraînés sur la base INRIA. Ces valeurs sont parmi les meilleures, seulement surpassées par **FPDW** qui arrive approximativement à 6.0 fps. Mais en fait **FPDW** repose sur les principes de **ChnFeats** et optimise le processus de détection en approximant les descripteurs le long d'un espace-échelle. Des techniques similaires pourraient être utilisées pour améliorer la vitesse de notre détecteur. D'un autre coté, le modèle entraîné sur la base de données Ladybug atteint un fps de 10.6 sur un jeu de données plus simple. C'est un avantage supplémentaire du fait que la majorité des m"ethodes de l'état de l'art n'ont pas la possibilité de changer automatiquement la complexité du détecteur entraîné sur un jeu de données, comme par exemple le détecteur HOG et le **HogLbp** qui ont une taille fixe de vecteur de descripteur quel que soit le jeu de données.

C.5 Conclusions et perspectives

Cet article présente un nouveau détecteur basé sur des descripteurs hétérogènes sélectionnés via un processus d'optimisation discrète sur leur performance et coût CPU conjointement. Le formalisme est validé sur la base publique d'images INRIA *i.e.*, les résultats sont conformes à nos attentes : le détecteur offre un excellent compromis entre performances et vitesse comparativement à la littérature.

Les travaux futurs portent sur une accélération supplémentaire du détection ainsi prototypé via son implémentation GPU (pour *Graphical Processing Units*) puis son intégration sur un robot mobile autonome.

Appendix D

Coopération entre un robot mobile et des caméras d'ambiance pour le suivi multi-personnes (publié dans RFIA 2012)

Résumé

Cet article décrit une stratégie de coopération entre des caméras d'ambiance et des capteurs embarqués sur un robot mobile pour : (i) suivre une personne donnée et identifiée par un tag/badge radio fréquence (RF), et (ii) faciliter sa navigation en présence de passants lors de l'exécution de cette mission. Nous privilégions une approche "tracking-by-detection" qui fusionne au sein d'un filtre particulaire par chaîne de Markov les détections visuelles déportées et les détections issues des divers capteurs embarqués (laser, vision active, RFID). Les performances du traqueur multi-personnes sont caractérisées par des évaluations qualitatives et quantitatives sur séquences pré-enregistrées. Enfin, l'intégration du système perceptuel sur le robot, le contrôle de ses actionneurs via des techniques d'asservissement visuel et diagramme d'espace libre au voisinage immédiat du robot, illustre la capacité du robot à suivre une personne donnée en espace humain encombré.

Mots Clef

Suivi multi-cibles, fusion de données, suivi Bayésien, réseaux de caméras, systèmes de perception coopérative

D.1 Introduction

Le déploiement de robots mobiles en environnement humain privatif mais aussi public répond à un enjeu sociétal majeur. Il s'agit à terme de voir ces robots interagir de façon naturelle et effective avec les humains partageant l'espace, notamment de poursuivre une personne donnée tout en évitant les passants durant l'exécution de cette mission. Cette tâche de poursuite est largement appréhendée dans la communauté Robotique [Calisi 2007, Chen 2007, Gockley 2007]. Néanmoins, l'aptitude à conjointement éviter les passants durant cette tâche de navigation reste peu explorée... ce qui est rédhibitoire à une interaction sûre et fiable lorsque le robot et les humains sont censés partager le même espace.

Certes, certains travaux à l'instar de [Calisi 2007] considérent les passants comme des obstacles statiques à éviter mais il nous semble opportun pour le robot de prendre en considération leur dynamique/trajectoire pour partager plus harmonieusement l'espace. Cette tâche de perception multi-personnes et contrôle associé du robot lors de sa navigation est autrement plus complexe à appréhender. Ainsi, le système perceptuel doit simultanément permettre : (i) de détecter, suivre et reconnaitre la personne cible *i.e.*, l'interlocuteur du robot parmi les autres passants, et (ii) de détecter/suivre les passants pour inférer leurs trajectoires au voisinage immédiat du robot. Les seules ressources embarquées nous semblent incompatibles avec ces diverses tâches. A l'instar de [Micheloni 2003], nous privilégions un système perceptuel qui associe caméras d'ambiance et perception embarquée pour tirer parti de champs de vue élargis qui vont induire une meilleure anticipation par le robot lors de ses déplacements et limiter les occultations.

Le but est, pour le robot, de poursuivre une personne donnée et identifiable par son badge RFID et son apparence vestimentaire tout en évitant les passants par des mouvements réactifs appropriés. Notre stratégie de perception repose à la fois sur des capteurs embarqués (vision monoculaire active, laser SICK 2D, lecteur RFID omnidirectionnel) et déportés *i.e.*, deux caméras d'ambiance tandis que nos fonctionnalités de détection, identification et suivi multi-personnes reposent sur un formalisme bayésien. Chakravarty *et al.* [Chakravarty 2009], Chia *et al.* [Chia 2009] ont certes mené des travaux similaires sur la coopération entre perception embarquée et déportée pour le suivi de personnes par un robot guide mais hélas sans tirer parti de la complémentarité des perceptions embarquée et déportée car les deux perceptions des personnes au voisinage du robot sont exclusives ; on parle alors d'approche décentralisée. Ici, notre stratégie vise à combiner celles-ci pour détecter et suivre distinctement toutes les personnes au voisinage immédiat du robot dans un seul et unique filtre agrégant les données sensorielles embarquées et déportées.

A ce titre, ces travaux exhibent deux contributions : (1) une stratégie de suivi multi-personnes combinant ces deux modes de perception dans un cadre probabiliste, (2) son intégration sur la plateforme robotique et le couplage avec les actionneurs du robot pour contrôler les mouvements du robot lors de sa navigation pour poursuivre un individu donné tout en évitant harmonieusement les passants.

L'article est structuré comme suit. La section D.2 décrit les diverses composantes de notre architecture perceptuelle. La détection de personnes et leur suivi sont formalisés en section D.3. La section D.4 présente des évaluations hors ligne *i.e.*, sur des séquences pré-enregistrées ; les résultats associés sont alors discutés. L'intégration de notre architecture complète sur le robot guide Rackham, et son exécution en ligne, sont présentées en section D.5. Enfin, la section D.6 récapitule nos contributions et ouvre sur quelques extensions envisagées pour ces travaux.

D.2 Description de notre architecture perceptuelle

D.2.1 Plateforme robotique

Notre système est composé d'un robot mobile et deux caméras couleur flea2 fixées sur les murs ; ces dernières disposent d'une résolution 640 × 480 pixels et sont connectées à un PC intel dual-core *via* une connexion firewire (figure D.1). Notre robot Rackham est une plateforme mobile type iRobot B21r ; il embarque divers capteurs *i.e.*, un laser SICK positionné à $38cm$ du sol et disposant d'un champ de vue de 180^o, une caméra numérique Micropix montée sur une platine site-azimut (PTU), et un lecteur de badges passifs RFID prototypé au laboratoire afin de détecter et identifier toute personne "taggée" au voisinage du robot *i.e.*, sur[0.5;4.5]m et 360^o [Germa 2010]. Rackham embarque également deux PCs (mono-CPU et bi-CPUs PIII à 850 MHz) et une connexion Ethernet sans fil. La figure D.1 illustre le système complet ; la communication entre le robot mobile et le PC dévolu aux caméras d'ambiance est assurée par wi-fi.

Enfin, l'architecture logicielle de Rackham est basée sur l'architecture LAAS **G**eno**M** [Alami 1998]. Elle s'articule autour de modules propres à chacune des fonctionnalités et développées en C/C++ .

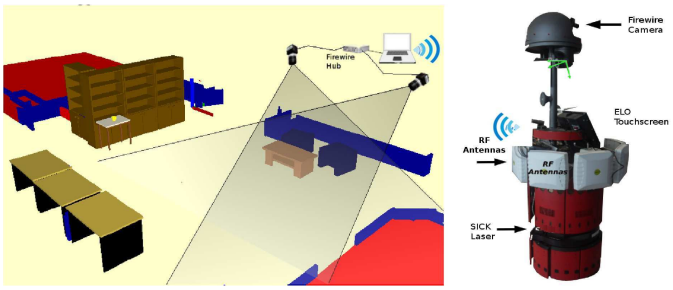


Figure D.1: Plateforme perceptuelle, caméras d'ambiance (positions et champs de vues associées) et robot mobile Rackham.

D.2.2 Synoptique descriptif du système

Le synoptique complet de notre architecture perceptuelle est présenté sur la figure D.2. Celle-ci s'articule autour de trois blocs/composantes dénommés A, B, and C. Le bloc A gère la détection et suivi d'une personne cible identifiable par son badge RFID. Ces fonctionnalités, développées lors de travaux antérieurs [Germa 2010], sont ici étendues afin de prendre en considération la détection de jambes par laser SICK et la détection visuelle par les caméras déportées.

Le bloc B, issu de développements récents, est dévolu à la perception des passants au voisinage immédiat du robot et constitue l'essence de cet article. Enfin, le bloc C gère les actionneurs du robot pour contrôler ses déplacements ; il repose sur les modalités précédentes afin à la fois de poursuivre la personne "taggée" tout en évitant les passants.

Les fonctionnalités perceptuelles précitées s'appuient sur divers modules de détection et suivi mono ou multi-personnes qui sont listés ci après.

(a) Modalités de détection

Diverses modalités de détection de personnes, dévolues aux capteurs pré-cités, sont mises en oeuvre ; celles-ci sont énumérées ici.

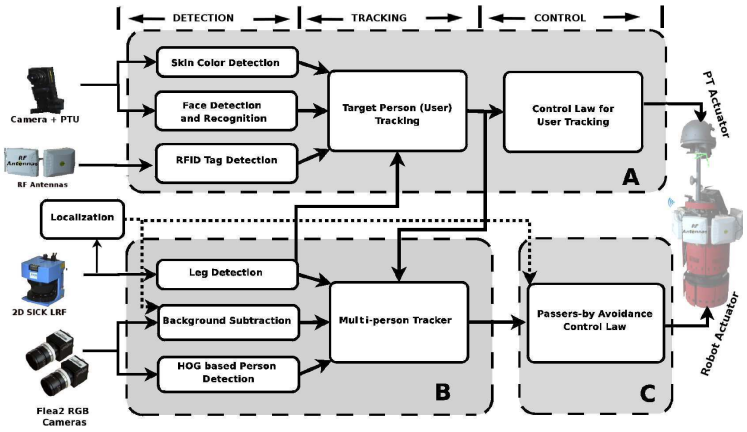
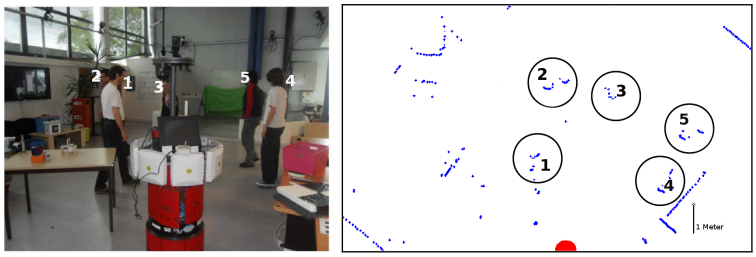


Figure D.2: Synoptique de notre architecture perceptuelle.

Laser SICK : Ce laser embarqué produit des coupes horizontales à $38cm$ du sol et sur 180^o avec une résolution angulaire de 0.5^o. La démarche vise alors à segmenter les jambes humaines grâce à des contraintes géométriques spécifiques ; le lecteur pourra se référer à [Xavier 2005] pour plus détails. La figure D.3 illustre un exemple de coupe laser SICK pour une situation homme-robot donnée ; les jambes segmentées sont matérialisées par des cercles.



(a) (b)

Figure D.3: Exemple de coupe laser SICK : situation homme-robot (a) et coupe laser associée (b). Les signatures correspondant à des jambes sont matérialisées par des cercles tandis que la position de Rackham est représentée par le cercle rouge.

Lecteur RFID : Nous avons prototypé et embarqué sur Rackham un lecteur RFID qui peut adresser jusqu'à 8 antennes et détecter tout tag passif sur 360^o et une distance $[0.5; 4.5]m$ relativement au robot. Le multiplexage de ces antennes par le lecteur permet de localiser grossièrement en distance et azimut le badge RF identifié. L'application suppose que l'interlocuteur du robot porte un tag RF acquis durant une phase préalable d'interaction proximale avec le robot. Ce badge facilite alors sa détection puis identification durant la phase de poursuite en présence de foule. Une illustration de détection RFID est montrée figure D.4(c).

Caméras d'ambiance : Les deux caméras fixées aux murs procurent des champs de vue larges et en recouvrement. Les personnes sont classiquement segmentées dans les flots vidéo associées par :

1. un algorithme type $\Sigma-\Delta$ [Manzanera 2007] segmentant les régions mobiles par soustraction de fond et respectant des proportions corporelles humaines. Une localisation extéroceptive par le système de vision binoculaire déporté, et supposé étalonné, permet d'exclure le robot mobile des zones mobiles résultantes.
2. un détecteur de personnes basé sur des histogrammes de gradients orientés (HOGs) à

(a) (b) (c)

Figure D.4: Détection RFID. Exemple de situation homme-robot (a), vue courante depuis la caméra embarquée et détection de visage associée (b). (c) illustre une carte de salience associée à la détection 360^o d'un badge RF donné. Le cercle rouge (resp. blanc) matérialise le robot Rackham (resp. le champ de vue du lecteur RFID sur $[0.5; 4.5]m$). Le lecteur RF pilote l'orientation azimutale de la caméra embarquée dont le champ de vue est représenté par un arc de cercle rouge.

l'instar de [Dalal 2005].

Ces deux détecteurs visuels sont illustrés sur la figure D.5. Chaque avatar (personne) détecté est alors caractérisé par sa position dans le plan du sol $(\mathsf{x}, \mathsf{y})_G$ et son apparence représentée par des histogrammes dans l'espace HSV.

Vision embarquée : Cette caméra couleur, montée sur platine site-azimut commandée par le système RFID, est dévolue à la perception proximale de la personne taggée. Elle s'appuie sur une détection/reconnaissance faciale et la détection de blobs peau, combinées à la détection RF, pour reconnaître et suivre la personne cible. La figure D.4-(b) montre un exemple de détection faciale depuis cette caméra.

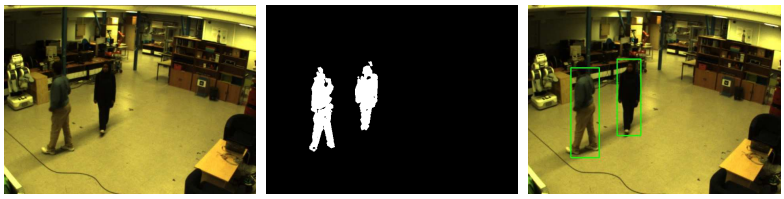
(b) Modalités de suivi

Suivi et poursuite de la personne "taggée" : Cette modalité gère l'identification de la personne "taggée", son suivi visuel dans le flot vidéo de la caméra embarquée, enfin sa poursuite par le robot. Ainsi, des techniques d'asservissement visuel sont mises en oeuvre pour : (i) piloter les actionneurs site-azimut de la caméra embarquée par le système RFID, et (ii) les actionneurs de la base mobile par le système complet RFID+vision. La fusion de données visuelles et RFID au sein d'un formalisme type Monte carlo, le contrôle du robot s'inscrivent dans des travaux antérieurs ; le lecteur pourra se référer à [Germa 2010] pour plus de détails. Une extension marginale a été ici d'enrichir la modalité par la détection laser. Enfin, la distribution colorimétrique de la personne "taggée" est considérée pour discriminer celle-ci des autres individus dans les caméras déportées.

Suivi des passants : Le traqueur multi-personnes tire parti des détecteurs pré-cités : laser SICK, vision déportée *i.e.*, segmentation des régions mobiles par soustraction de fond et détection de personnes par HOG pour alimenter le processus de suivi. Toute détection est projetée sur le sol $(\mathsf{x}, \mathsf{y})_G$ qui est supposé planaire et calibré comme les caméras déportées. A l'instar de [Khan 2003], nous privilégions un formalisme type filtre particulaire à chaîne de Markov avec sauts réversibles (RJMCMC-PF) pour gérer un nombre variable a priori de cibles/passants au voisinage du robot. Notre stratégie est détaillée dans la section suivante.

D.3 Modalité de suivi de passants

Notre but est de suivre simultanément et de façon robuste plusieurs personnes dans le champ de vue des capteurs et d'obtenir leurs trajectoires au cours de leurs déplacements. Plusieurs approches sont proposées dans la littérature. Le filtrage particulaire avec Chaîne de Markov et sauts réversibles (noté RJMCMC – PF)) est très adapté pour suivre des cibles en intéraction



(a) (b) (c)

Figure D.5: Exemple d'images acquise par les caméras déportées : champ de vue (a), segmentation des régions mobiles par $\Delta - \Sigma$ (b), détection de personnes par HOG (c).

comme illustré dans [Khan 2003]. Nous avons donc adapté ce formalisme à notre stratégie de perception coopérative qui repose sur des capteurs hétérogènes embarqués et déportés.

D.3.1 Formalisme RJMCMC – PF

Le formalisme RJMCMC – PF remplace l'étape de rééchantillonnage par importance par une étape d'échantillonnage RJMCMC. La distribution *a posteriori*, $P(X_t|Z_{1:t})$, est approximée par un ensemble d'échantillons non pondérés, $P(X_t|Z_{1:t}) \approx \frac{1}{N}\sum_{n=1}^{N}\delta(X_t - X_t^n)$, où X_t^n représente la n^{th} particule, dans un cadre bayésien. L'état d'une particule caractérise la configuration de l'ensemble des cibles suivies : $X_t^n = \{I_t^n, x_{(t,i)}^n\}, i \in \{1, ..., I_t^n\}$, où I_t^n est le nombre d'objets suivis de l'hypothèse n au temps t, et $x_{(t,i)}^n$ est un vecteur représentant l'état de l'objet i. L'estimation *a posteriori* est réalisée en définissant une chaîne de Markov sur l'espace des configurations à dimension variable X_t^n de telle façon que la distribution stationnaire de la chaîne soit égale à la probabilité *a posteriori* désirée. Ainsi, à chaque instant, le filtre commence une chaîne de Markov à partir d'une configuration initiale et effectue $N + N_B$ itérations, où N est le nombre de particules et N_B représente le nombre d'itérations nécessaires pour converger vers les échantillons stationnaires, proposant une nouvelle hypothèse, X^*, à partir de l'hypothèse précédente, X, dépendant du mouvement choisi qui, soit modifie soit laisse inchangée la dimension de l'état. Les N particules finales représentent une approximation de la distribution *a posteriori* recherchée.

D.3.2 Implémentation

Notre traqueur RJMCMC – PF vise à caractériser la trajectoire des cibles dans le plan du sol et leurs apparences associées. Nos choix quant au vecteur d'état estimé, les modèles de dynamique, d'interaction et d'observation relatifs au filtre RJMCMC – PF sont discutés ci-après.

D.3.2.1 Vecteur d'état

Le vecteur d'état d'une cible i pour l'hypothèse n au temps t est un vecteur contenant l'identifiant Id et la position (x, y) sur le plan du sol dans un référentiel associé $x_{t,i}^n = \{Id_i, \mathsf{x}_{t,i}^n, \mathsf{y}_{t,i}^n\}$.

D.3.2.2 Mouvements proposés

Le filtre RJMCMC – PF gère un espace d'état de dimension variable. L'espace d'état agrège plusieurs sous-espaces relatifs aux diverses cibles couramment suivies avec éventuellement des sauts vers des espaces supérieurs ou inférieurs selon les entrées/sorties de ces cibles dans le voisinage du robot. Les mouvements proposés permettent à chaque itération de guider l'exploration de cet espace d'état. Dans notre implémentation, nous proposons un ensemble de quatre mouvements possibles, $m = \{Ajout, MiseAJour, Suppression, Permutation\}$. Le choix du mouvement à chaque itération est déterminé par q_m, la distribution du mouvement. Ces valeurs sont

déterminées empiriquement et valent $\{0.15, 0.02, 0.8, 0.03\}$. Pour simplifier à la fois la transition entre la nouvelle hypothèse d'état proposée X^* et X, et l'évaluation du taux d'acceptation, nous ne considérons que des changements vers un sous-ensemble aléatoire de l'espace (dans le cas d'un suivi multi-cibles, cela signifie qu'une seule cible est modifiée par itération). L'équation D.1 présente le calcul du taux d'acceptation d'une proposition X^*. Le terme q_m est la probabilité que le mouvement m soit proposé, $Q_m(X^{t-1}|X^*)$ est la probabilité de générer X^{t-1} en perturbant X^* par le mouvement m. Le terme $\pi(X_t^n)$ est le modèle d'observation et $\Psi(X_t^n)$ est le modèle d'interaction; ceux-ci sont détaillés ci-après.

$$\beta = min(1, \frac{\pi(X^*)Q_{m*}(X_t^{n-1}|X^*)q_{m*}\Psi(X^*)}{\pi(X_t^{n-1})Q_m(X^*|X_t^{n-1})q_m\Psi(X_t^{n-1})}) \tag{D.1}$$

où $m \in \{Ajout, MiseAJour, Suppression, Permutation\}$ de cibles et m^* est le mouvement inverse de m.

Mouvement "Ajout" cible: Le mouvement $Ajout$ choisit de façon aléatoire une personne détectée, x_p, et ajoute son vecteur d'état à X_t^{n-1} ce qui résulte en l'état proposé X^*. La densité contrôlant le mouvement $Ajout$, $Q_{Ajout}(X^*|X_t^{n-1})$, est alors calculée par l'équation D.2. On construit alors une carte de probabilités, composée de mixtures de gaussiennes associées aux cibles détectées, chaque détection étant représentée par une gaussienne sur le plan du sol, et cette carte est modifiée d'après les cibles suivies $\hat{X}$ à l'instant $t-1$ de telle façon que la distribution aura de plus fortes valeurs aux endroits où des cibles détectées ne sont pas encore suivies.

$$Q_{Ajout}(X^*|X_t^{n-1}) = \sum_d k_d . \sum_{j=1}^{N_d} \mathcal{N}(x_p; z_{t,j}^d, \Sigma).(1 - k_m \sum_{j=1}^{N_t} \mathcal{N}(x_p; \hat{X}_{t-1,j}, \Sigma)) \tag{D.2}$$

où d représente l'ensemble des détecteurs $\{l, sc_1, sc_2\}$, N_d est le nombre total de détections de chaque détecteur, k_d est une pondération associée à chaque détecteur telle que $\sum_d k_d = 1$, $\hat{X}_{t-1}$ est l'estimée par MAP du filtre à l'instant $t-1$, N_t est le nombre total de cibles dans l'estimation par MAP et k_m est une constante de normalisation.

Mouvement "Suppression" cible: Le mouvement $Suppression$ choisit aléatoirement une personne suivie, x_p, de la particule considérée, X_t^{n-1}, et la supprime pour proposer un nouvel état X^*. Contrairement au mouvement $Ajout$, la densité de probabilités proposée utilisée quand on calcule le taux d'acceptance, $Q_{Suppression}(X^*|X_t^{n-1})$ (equation D.3), est donnée par la carte de distribution issue des cibles suivies modifiée par les cibles détectées. Cette carte favorise les suppressions de cibles qui sont sorties de la zone de suivi mais qui sont toujours suivies.

$$Q_{Suppression}(X^*|X_t^{n-1}) = (1 - \sum_d k_d . \sum_{j=1}^{N_d} \mathcal{N}(x_p; z_{t,j}^d, \Sigma)).(k_m \sum_{j=1}^{N_t} \mathcal{N}(x_p; \hat{X}_{t-1,j}, \Sigma)) \tag{D.3}$$

Mouvement "Mise à jour" cible: Dans le mouvement $Misejour$, le vecteur d'état d'une cible choisie aléatoirement est perturbé par une distribution normale. La densité de probabilité, $Q_{MiseAJour}(X^*|X_t^{n-1})$, est alors une distribution normale avec comme moyenne la position de la cible mise à jour. Ici le taux d'acceptation est influencé par l'évaluation de la vraisemblance et les interactions entre les cibles.

Mouvement "Permutation" cible: Le mouvement *Permutation* permet d'échanger les identifiants de cibles proches ou en interaction. Quand ce mouvement est sélectionné, les identifiants de deux des plus proches cibles sont échangés et une nouvelle hypothèse X^* est proposée. Le taux d'acceptation est alors calculé de la même façon que pour le mouvement $MiseAJour$.

D.3.2.3 Modèle d'interaction

Pour représenter l'interaction entre cibles, nous introduisons un potentiel d'interaction entre cibles susceptibles d'interagir donc proches spatialement. Par souci de simplicité, ce potentiel est limité entre deux cibles mais ce modèle pourrait être étendu à un nombre quelconque de cibles. La finalité est de maintenir autant que possible une identité par traqueur et pénaliser les configurations associant deux traqueurs à une seule cible. A l'instar de [Khan 2003, Smith 2005], nous privilégions des champs aléatoires de Markov (MRF pour *Markov Random Field*) pour modéliser ces interactions. Ainsi, pour un état donné X_t^n, notre modèle MRF relatif à un état donné X_t^n s'exprime par l'équation D.4.

$$\begin{aligned} \Psi(X_t^n) &= \Pi_{i\neq j}\phi(x_{t,i}^n, x_{t,j}^n) \\ \phi(x_{t,i}^n, x_{t,j}^n) &= 1 - exp(-(\frac{d(x_{t,i}^n, x_{t,j}^n)}{\sigma})^2) \end{aligned} \tag{D.4}$$

où $d(x_{t,i}^n, x_{t,j}^n)$ est la distance euclidienne, $i, j \in \{1, ..., I_t^n\}$, et I_t^n est le nombre de cibles dans X_t^n.

D.3.2.4 Modèle d'observation

Le modèle d'observation s'appuie sur les détections, excepté pour le laser pour lequel on considère les blobs formés d'après les données brutes du laser, et si le mouvement proposé est $Misejour$ ou $Permutation$, c'est alors une mesure de Bhattacharyya. Les données brutes du laser sont filtrées de façon à créer des blobs et à ne garder que ceux qui ont une forme assimilable à une paire de jambes; ces blobs sont notés l_b. Ceci permet alors de filtrer les éléments comme les murs, les tables, les pieds de chaise ou autres. Les blobs laser ainsi retenus et les blobs associés aux détections visuelles sont représentés par une mixture de gaussiennes exprimée dans le plan du sol et centrée sur les zones de détection. En annotant z_t toute mesure extéroceptive acquise à l'instant t; le modèle d'observation pour la n^{th} particule X_t^n est alors calculé d'après l'équation D.5.

$$\pi(X_t^n) = \pi_B(X_t^n).\pi_D(X_t^n) \tag{D.5}$$

$$\pi_B(X_t^n) = \begin{cases} \prod_{i=1}^{M}\prod_{c=1}^{2} e^{-\lambda B_{i,c}^2}, & \text{si } MiseAJour \text{ ou } Permutation \\ 1 & , \quad \text{sinon} \end{cases}$$

$$\pi_D(X_t^n) = \frac{1}{M}\sum_{i=1}^{M}(\sum_d k_d.\pi(x_i|z_t^d)), \sum_d k_d = 1$$

$$\pi(x_i, z_t^d) = \frac{1}{N_d}\sum_{j=1}^{N_d}\mathcal{N}(x_i; z_{t,j}^d, \Sigma)$$

Dans l'équation D.5, B_i représente la distance de Bhattacharyya calculée entre l'histogramme

d'apparence de la cible proposée i dans la particule X_t^n et le modèle de cible pour chaque caméra c. M représente le nombre de cibles pour la particule, et N_d est le nombre total de détections pour chaque modalité vision ou laser *i.e.*, d, $d = \{l_b, c_1, c_2\}$. Enfin, k_d est un poids assigné à chaque détecteur et fonction de leur précision respective tandis que x_i représente la position de la cible i dans le plan du sol.

D.4 Évaluations

Les performances de notre traqueur RJMCMC – PF sont évaluées sur trois séquences vidéos acquises simultanément depuis les capteurs embarqués sur Rackham et les caméras déportées. La scène couverte par la plateforme perceptuelle est approximativement 10x8.20m^2; le robot Rackham navigue librement dans ce lieu, ici un espace public partagé avec les passants. La séquence I (resp. II) contient 200 images avec présence simultanée de deux (resp. trois) cibles/passants au voisinage du robot. Enfin, la séquence III, comportant 185 images, inclut ici quatre cibles mobiles.

Les performances sont évaluées via les critères suivants [Bardet 2009] :

- Taux de Réussite du Suivi (TRS): donné par $\frac{1}{J_t}\sum_{k,j}\delta_{k,j}$ où $\delta_{k,j} = 1$ si la cible j est suivie à l'instant t, sinon 0. $J_t = \sum_{k,j} j_k$, et j_k représentent le nombre de personnes dans la zone de suivi à l'image k.

- Nombre Moyen de faux positifs par image (MFP): comptabilise le nombre de faux positifs par image, *i.e.*, $\frac{1}{K}\sum_{k,j}\delta_{k,j}$ avec $\delta_{k,j} = 1$ si la cible suivie j n'est pas une cible à l'image k, sinon 0. K est le nombre total d'images dans la base de données.

- Taux de Fantômes (TF): calcule le nombre de cibles candidates sur le nombre de non-cibles (fantômes) en moyenne sur le nombre total de cibles dans la base de données, *i.e.* $\frac{1}{J_t}\sum_{k,j}\delta_{k,j}$ avec $\delta_{k,j} = 1$ si la cible suivie j est un fantôme à l'image k, sinon 0.

- Erreur Moyenne de Précision (EMP): mesure la précision avec laquelle les cibles sont suivies comme étant la somme des erreurs quadratiques entre la position suivie estimée et la vérité terrain, en moyenne sur la séquence totale. Elle est exprimée en centimètres (cm).

- Saut d'identifiant (SID) : ce critère quantifie combien de fois un identifiant "saute" entre deux cibles suivies. Il est représenté comme étant $\sum_k \sum_{i,j} \delta_{i,j}$, où $\delta_{i,j} = 1$ quand un saut d'identifiant a lieu entre la cible suivie i et la cible j à l'image k, sinon cela vaut 0,

Pour chaque séquence, une vérité terrain labellisée à la main avec la position réelle (x, y) et un identifiant unique par personne, est utilisée. Une personne est considérée comme suivie correctement si la position suivie est à 30cm au maximum de la vérité terrain. Chaque séquence est exécutée huit fois pour tenir compte de la nature stochastique du filtre. Les résultats présentés dans le tableau D.1 représentent les moyennes et écart-types associés. Pour quantifier l'apport de la seconde caméra (c_2), les performances du traqueur couplant le laser à une seule des deux caméras déportées (c_1) sont aussi mentionnés.

D'après les résultats, nous pouvons observer que le traqueur donne de bons résultats avec un bon taux de succès même quand il y a simultanément quatre personnes dans la même image. L'ajout de la seconde caméra améliore les performances du traqueur vis-à-vis de tous les critères utilisés. La figure D.6 montre quelques *snapshots* du tracking sur la séquence II. Ces images illustrent l'initialisation du traqueur sur une nouvelle cible détectée (D.6a)(chemin rouge), sa ré-initialisation après échec - trajet bleu puis noir sur la figure (D.6b), la complémentarité des

Table D.1: Résultats d'évaluation du suivi (moyenne et écart-type).

	Traqueur	Seq. I	Seq. II	Seq. III
TRS	avec c_1	0.932 ± 0.01	0.856 ± 0.04	0.72 ± 0.03
	avec c_1, c_2	0.934 ± 0.03	0.886 ± 0.02	0.750 ± 0.02
MFP	avec c_1	0.376 ± 0.07	0.186 ± 0.05	1.018 ± 0.18
	avec c_1, c_2	0.23 ± 0.05	0.156 ± 0.08	0.63 ± 0.28
TF	avec c_1	0.21 ± 0.04	0.07 ± 0.02	0.31 ± 0.05
	avec c_1, c_2	0.132 ± 0.03	0.058 ± 0.03	0.168 ± 0.10
$\text{EMP}_{(\text{cms})}$	avec c_1	17.85 ± 0.447	17.63 ± 0.96	23.53 ± 1.23
	avec c_1, c_2	17.63 ± 0.96	17.456 ± 2.25	20.48 ± 0.79
SID	avec c_1	0	0.8 ± 0.84	1.4 ± 1.26
	avec c_1, c_2	0	0.2 ± 0.45	1.2 ± 1.03

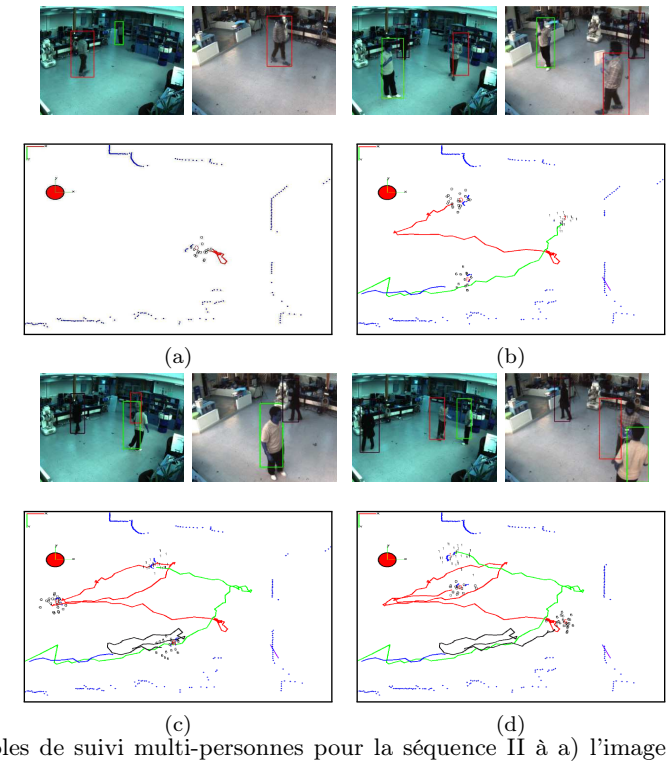


Figure D.6: Exemples de suivi multi-personnes pour la séquence II à a) l'image 27, b) l'image 60, c) l'image 94, et d) l'image 104. Les deux images du haut correspondant au flux de la caméra et celle du bas montre le plan du sol avec les trajectoires des personnes suivis en surimposition. Le nuage de particules est aussi présenté avec les identifiants de chaque individu. Les points bleus correspondent aux impacts laser.

capteurs pour gérer au mieux les occultations partielles (D.6c), et le suivi pendant une interaction (D.6d). [1]

[1] Le lecteur pourra se référer au lien URL homepages.laas.fr/aamekonn/videos/ pour visionner les séquences complètes.

D.5 Intégration sur le robot Rackham et démonstration

Les fonctionnalités précédentes sont implémentées en C/C++ et embarquées sur Rackham via des modules **G**enoM dédiés. La communication sans fil entre CPU dédiée aux caméras déportées et CPU embarquée sur le robot est faible débit et doit donc éviter tout transfert d'images brutes. Ainsi, la détection des passants est réalisée sur la CPU déportée tandis que les autres fonctionnalités sont gérées par la CPU embarquée sur Rackham.

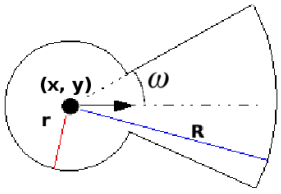
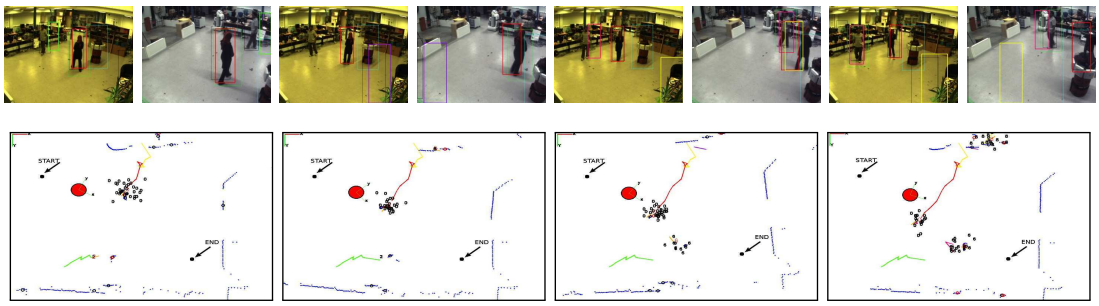


Figure D.7: Zone de sécurité autour d'une cible et définie sur le plan du sol. Elle est paramétrée par r, R et ω, le point noir représentant la position de la personne. La flèche décrit le cap de la cible estimé par le traqueur.

Le traqueur multi-personnes s'exécute à environ 1 fps à l'instar de nos évaluations hors-ligne. Concernant la loi de commande d'évitement des passants, une zone de sécurité autour de chaque cible est définie (figure D.7) en tirant parti de sa trajectoire estimée. Cette zone ici paramétrée par (r, R, ω) vise à éviter de couper les trajectoires des passants et ainsi partager l'espace harmonieusement. Ainsi, la navigation du robot repose sur la coopération entre deux lois de commande pour simultanément : (i) induire des mouvements réactifs compatible avec la carte d'espace libre au voisinage immédiat du robot, (ii) poursuivre la cible "taggée". Le lecteur pourra se référer à [Minguez 2004] pour plus de détails sur la formalisation de cette carte d'espace libre.



(a) (b) (c) (d)

Figure D.8: Illustration du comportement de Rackham en présence de plusieurs passants. Zone de sécurité paramétrée par $r = 0.5m$, $R = 1.5m$ et $\beta = 30^0$.

Pour valider les fonctionnalités ainsi intégrées, le scénario vise à faire naviguer Rackham entre deux points prédéfinis A et B, tout en évitant deux passants... et ceci sur plusieurs *runs*. La figure D.8 montre des illustrations prises depuis le module de suivi, montrant la position du robot, la trajectoire des passants, et le point de départ et d'arrivée du déplacement du robot. Les séquences illustrent clairement les mouvements adaptés du robot tourne pour passer derrière le passant percu, en couleur rouge.[2]

Finalement, nous avons évalué (qualitativement) le système complet, *i.e.*, les diverses fonctionnalités de détection/suivi:reconnaissance de la personne "taggée", la détection/suivi des

[2]Le lecteur pourra se référer au lien URL homepages.laas.fr/aamekonn/videos/ pour visionner les séquences complètes.

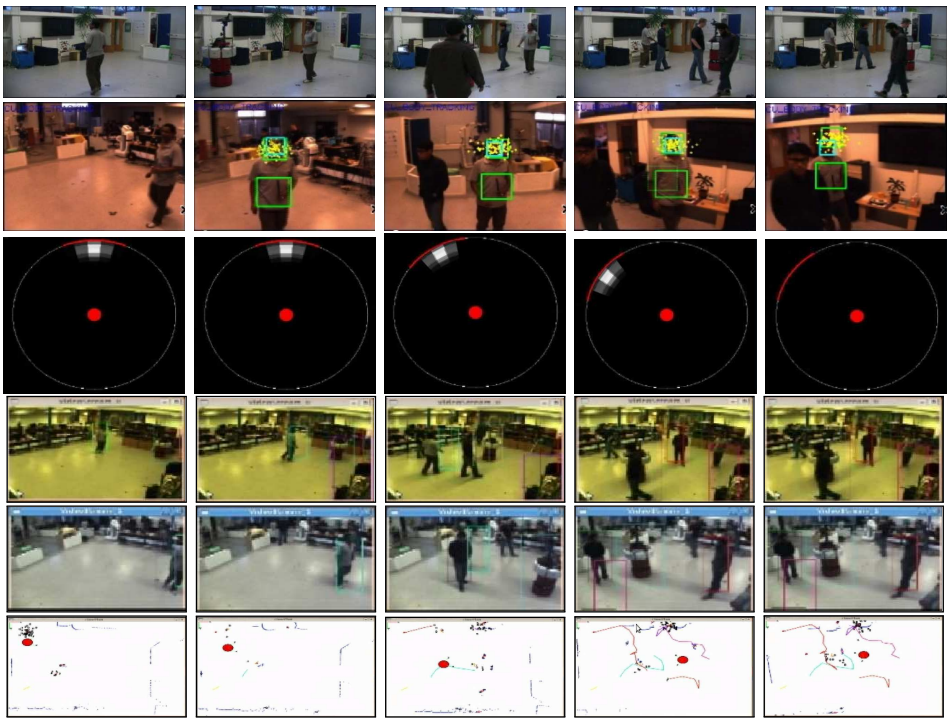
passants mobiles mais aussi les lois de commande associées sur un scénario incluant 5 personnes gravitant au voisinage du robot. Les vidéos acquises pour ce scénario sont illustrées figure D.9.[2]

D.6 Conclusion et Perspectives

La navigation d'un robot en présence de "foule" est un enjeu sociétal majeur dans la perspective de voir des robots interactifs partager harmonieusement la tâche et donc l'espace avec les usagers humains du lieu. La perception simultanée de l'interlocuteur du robot mais aussi des passants est alors vitale. Notre contribution propose un schéma de coopération entre des caméras d'ambiance et des capteurs embarqués sur un robot mobile. La fusion probabiliste de données sensorielles hétérogènes issues de capteurs déportées et embarquées permet de surpasser (1) les systèmes classiques de surveillance basés uniquement sur des caméras fixes qui ne peuvent pas gérer d'angle mort, et (2) les systèmes complètement embarqués sans large champ de vue et avec une capacité simpliste de (re)-initialisation. Ce constat est validé par les résultats hors-ligne exhibés par notre traqueur RJMCMC − PF. Ce dernier offre à notre robot Rackham la capacité de différencier une personne cible/"taggée dans une "foule" (relative...), poursuivre ce dernier tout en prenant en considération la dynamique des autres humains pour les éviter harmonieusement.

La plateforme perceptuelle a été validée sur le robot Rackham après intégration et évaluations sur des scénarios robotiques mettant en jeu deux à cinq personnnes au voisinage du robot. Les évaluations, certes qualitatives, illustrent le comportement satisfaisant du robot pour ces divers scénarios puisque ce dernier est capable de suivre de façon robuste une personne donnée tout en évitant les passants.

Les investigations futures visent à mener des évaluations quantitatives en ligne et pour des environnements encore plus encombrés. La gestion de plusieurs tags est également à explorer pour autoriser une interaction avec plusieurs humains simultanément. Enfin, nous allons considérer des caméras externes PTZ afin de (1) accroître le champ de vue, et (2) gérer les ressources perceptuelles de façon optimale *e.g.*, dédié un flux vidéo à une personne cible donnée.



(a) 4.7*sec* (b) 10.8*sec* (c) 15.9*sec* (d) 39.7*sec* (e) 40.6*sec*

Figure D.9: Illustrations issues de la poursuite d'utilisateur en évitant les passants. Chaque ligne correspond au flux issu de : la caméra externe capturant la situation H/R, la vision embarquée, la détection RFID, les deux caméras déportées et le plan du sol montrant les trajectoires de suivi et le balayage laser. L'instant auquel sont capturées les images est indiqué, En (a), l'utilisateur est entièrement visible dans la scène, avec la détection RF indiquant la direction du tag portée par l'utilisateur. Le suivi de l'utilisateur peut être vu en (b) avec les particules jaunes dans le flux de la caméra mobile, et une boite englobante verte dans le flux des caméras d'ambiance. En fait le robot part pour suivre l'utilisateur suivant un chemin rectiligne car il n'y a pas de passant à proximité. Des passants apparaissent en (c) et leurs suivis de trajectoirent peuvent être observés sur le plan du sol montrant le balayage laser. (d)-(f) montrent le suivi et les évitements pendant les expérimentations.

Bibliography

[Alami 1998] R. Alami, R. Chatila, S. Fleury, M. Ghallab and F. Ingrand. *An Architecture for Autonomy*. International Journal of Robotics Research, vol. 17, pages 315–337, 1998.

[Alexe 2010] B. Alexe, T. Deselaers and V. Ferrari. *What is an Object?* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), San Francisco, CA, USA, June 2010.

[ALGLIB] Sergey Bochkanov. *ALGLIB*. (www.alglib.net).

[Arampatzis 2005] T. Arampatzis, J. Lygeros and S. Manesis. *A Survey of Applications of Wireless Sensors and Wireless Sensor Networks*. In IEEE International Symposium on Intelligent Control (ISIC'05), Limassol,Cyprus, June 2005.

[Arras 2007] K. O. Arras, Ò. Martínez and M. W. Burgard. *Using Boosted Features for Detection of People in 2D Range Scans*. In International Conference on Robotics and Automation (ICRA'07), Rome, Italy, April 2007.

[Arras 2012] K.O. Arras, B. Lau, S. Grzonka, M. Luber, O.M. Mozos, D. Meyer-Delius and W. Burgard. *Range-Based People Detection and Tracking for Socially Enabled Service Robots*. In Erwin Prassler, Marius Zöllner, Rainer Bischoff, Wolfram Burgard, Robert Haschke, Martin Hägele, Gisbert Lawitzky, Bernhard Nebel, Paul Plöger and Ulrich Reiser, editeurs, Towards Service Robots for Everyday Environments, volume 76 of *Springer Tracts in Advanced Robotics*, pages 235–280. Springer Berlin Heidelberg, 2012.

[Arsic 2008] D. Arsic, E. Hristov, N. Lehment, B. Hornler, B. Schuller and G. Rigoll. *Applying Multi Layer Homography for Multi Camera Person Tracking*. In International Conference on Distributed Smart Cameras (ICDSC'08), Stanford, CA, USA, September 2008.

[Bardet 2009] F. Bardet, T. Chateau and D. Ramadasan. *Illumination aware MCMC Particle Filter for Long-term Outdoor Multi-object Simultaneous Tracking and Classification*. In IEEE International Conference on Computer Vision (ICCV'09), Kyoto, Japan, October 2009.

[Beauchemin 1995] S. S. Beauchemin and J. L. Barron. *The Computation of Optical Flow*. ACM Computing Surveys, vol. 27, no. 3, pages 433–466, September 1995.

[Bellotto 2009] N. Bellotto and H. Hu. *Multisensor-Based Human Detection and Tracking for Mobile Service Robots*. IEEE Transactions on Systems, Man, and Cybernetics – Part B, vol. 39, no. 1, pages 167–181, 2009.

[Belongie 2002] S. Belongie, J. Malik and J. Puzicha. *Shape Matching and Object Recognition using Shape Contexts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pages 509–522, 2002.

[Bennewitz 2005] M. Bennewitz, F. Faber, D. Joho, M. Schreiber and S. Behnke. *Towards a Humanoid Museum Guide Robot that Interacts with Multiple Persons*. In IEEE-RAS International Conference on Humanoid Robots (Humanoids'05), Tsukuba, Japan, 2005.

[Bernardin 2008] K. Bernardin and R. Stiefelhagen. *Evaluating multiple object tracking performance: the CLEAR MOT metrics*. EURASIP Journal on Image and Video Processing, vol. 2008, pages 1:1–1:10, January 2008.

[Beyan 2012] C. Beyan and A. Temizel. *Adaptive mean-shift for automated multi object tracking*. Computer Vision, IET, vol. 6, no. 1, pages 1–12, 2012.

[Beymer 2002] D. Beymer and K. Konolige. *Tracking People from a Mobile Platform*. In International Symposium on Experimental Robotics (ISER'02), Sant'Angelo d'Ischia, Italy, Italy 2002.

[Bouma 2013] H. Bouma, J. Baan, S. Landsmeer, C. Kruszynski, G. Antwerpen and J. Dijk. *Real-time Tracking and Fast Retrieval of Persons in Multiple Surveillance Cameras of a Shopping Mall*. In Proceedings of SPIE, May 2013.

[Breitenstein 2009] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier and L. Van Gool. *Robust tracking-by-detection using a detector confidence particle filter*. In International Conference on Computer Vision (ICCV'09), Kyoto, Japan, October 2009.

[Breitenstein 2011] M.D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier and L. Van Gool. *Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 33, no. 9, pages 1820–1833, 2011.

[Broggi 2000] A. Broggi, M. Bertozzi, A. Fascioli and M. Sechi. *Shape-based Pedestrian Detection*. In IEEE Intelligent Vehicles Symposium (IV'00), Dearborn, MI, USA, October 2000.

[Broggi 2006] A. Broggi, R.L. Fedriga and A. Tagliati. *Pedestrian Detection on a Moving Vehicle: an Investigation about Near Infra-Red Images*. In IEEE Intelligent Vehicles Symposium (IV'06), Tokyo, Japan, June 2006.

[Brückmann 2006] R. Brückmann, A. Scheidig, C. Martin and H.-M. Gross. *Integration of a Sound Source Detection into a Probabilistic-based Multimodal Approach for Person Detection and Tracking*. In P. Levi, M. Schanz, R. Lafrenz and V. Avrutin, editeurs, Autonome Mobile Systeme 2005, Informatik aktuell, pages 131–137. Springer Berlin Heidelberg, 2006.

[Cadenat 2012] V. Cadenat, D. Folio and A.D. Petiteville. *Comparison of Two Sequencing Techniques to Perform a Vision-Based Navigation Task in a Cluttered Environment*. Advanced Robotics, vol. 26, no. 5-6, pages 487–514, 2012.

[Calisi 2007] D. Calisi, L. Iocchi and G. R. Leone. *Person Following through Appearance Models and Stereo Vision using a Mobile Robot*. In International Workshop on Robot Vision, Barcelona, Spain, March, 2007.

[Carballo 2009] A. Carballo, A. Ohya and S. Yuta. *People Detection using Double Layered Multiple Laser Range Finders by a Companion Robot*. In H. Hahn, H. Ko and S. Lee, editeurs, Multisensor Fusion and Integration for Intelligent Systems, volume 35 of *Lecture Notes in Electrical Engineering*, pages 315–331. Springer Berlin / Heidelberg, 2009.

[Chakravarty 2006] P. Chakravarty and R. Jarvis. *Panoramic Vision and Laser Range Finder Fusion for Multiple Person Tracking*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06), Beijing, China, October 2006.

[Chakravarty 2009] P. Chakravarty and R. Jarvis. *External Cameras and A Mobile Robot: A Collaborative Surveillance System*. In Australasian Conference on Robotics and Automation (ACRA'09), Sydney, Australia, December 2009.

[Chen 2002] L. Chen, P.O Arambel and R.K. Mehra. *Estimation under Unknown Correlation: Covariance Intersection Revisited*. IEEE Transactions on Automatic Control, vol. 47, no. 11, pages 1879–1882, 2002.

[Chen 2003] Z. Chen. *Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond*. Technical report, McMaster University, 2003.

[Chen 2007] Z. Chen and S. T. Birchfield. *Person Following with a Mobile Robot Using Binocular Feature-Based Tracking*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07), Sand Diego, CA, USA, November 2007.

[Chen 2008] C.-H. Chen, Y. Yao, D. Page, B. Abidi, A. Koschan and M. Abidi. *Heterogeneous Fusion of Omnidirectional and PTZ Cameras for Multiple Object Tracking*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 8, pages 1052–1063, 2008.

[Chia 2009] C.C. Chia, W.K. Chan and S.Y. Chien. *Cooperative Surveillance System with Fixed Camera Object Localization and Mobile Robot Target Tracking*. In T. Wada, F. Huang and S. Lin, editeurs, Advances in Image and Video Technology, volume 5414, pages 886–897. Springer Berlin / Heidelberg, 2009.

[Choi 2011] W. Choi, C. Pantofaru and S. Savarese. *Detecting and tracking people using an RGB-D camera via multiple detector fusion*. In ICCV Workshops, Barcelona, Spain, November 2011.

[Choi 2013] W. Choi, C. Pantofaru and S. Savarese. *A General Framework for Tracking Multiple People from a Moving Camera*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 7, pages 1577–1591, 2013.

[Chong 2008] E. K. P. Chong and S. H. Zak. *Multi-objective Optimization*. In An Introduction to Optimization, Third Edition, pages 541–563. John Wiley & Sons, Inc., 2008.

[Clodic 2006] A. Clodic, S. Fleury, R. Alami, R. Chatila, G. Bailly, L. Bréthes, M. Cottret, P. Danès, X. Dollat, F. Eliseï, I. Ferrané, M. Herrb, G. Infantes, C. Lemaire, F. Lerasle, J. Manhes, P. Marcoul, P. Menezes and V. Montreuil. *Rackham: An Interactive Robot-Guide*. In International Conference on Robot-Machine Interaction (ROMAN'06), Hatfield, UK, September 2006.

[Cnet 2005] Cnet News. *Robot guard for the office*, 2005. http://news.cnet.com/2300-7355_3-5759293.html, retrieved: Novemeber 19, 2013.

[Correa 2012] M. Correa, G. Hermosilla, R. Verschae and Javier Ruiz-del Solar. *Human Detection and Identification by Robots Using Thermal and Visual Information in Domestic Environments*. Journal of Intelligent and Robotic Systems, vol. 66, pages 223–243, 2012.

[Cory 1999] P. Cory, H. R. Everett and T. A. Heath-Pastore. *Radar-based Intruder Detection for a Robotic Security System*. In Proceedings of SPIE, volume 3525, January 1999.

[Cui 2005] J. Cui, H. Zha, H. Zhao and R. Shibasaki. *Tracking Multiple People Using Laser and Vision*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05), Edmonton, Canada, August 2005.

[Cui 2008] J. Cui, H. Zha, H. Zhao and R. Shibasaki. *Multi-modal Tracking of People using Laser Scanners and Video Camera*. Image and Vision Computing, vol. 26, no. 2, pages 240 – 252, 2008.

[Dalal 2005] N. Dalal and B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, June 2005.

[Dalal 2006a] N. Dalal. *Finding People in Images and Videos*. PhD thesis, Institut National Polytechnique de Grenoble, July 2006.

[Dalal 2006b] N. Dalal, B. Triggs and C. Schmid. *Human Detection Using Oriented Histograms of Flow and Appearance*. In European Conference on Computer Vision (ECCV'06), Graz, Austria, May 2006.

[Di Paola 2010] D. Di Paola, A. Milella, G. Cicirelli and A. Distante. *An Autonomous Mobile Robotic System For Surveillance Of Indoor Environments*. International Journal of Advanced Robotic Systems, vol. 7, no. 1, pages 19–26, March 2010.

[Dollár 2008] P. Dollár, B. Babenko, S. Belongie, P. Perona and Z. Tu. *Multiple Component Learning for Object Detection*. In European Conference on Computer Vision (ECCV'08), Marseille, France, October 2008.

[Dollár 2009] P. Dollár, Z. Tu, P. Perona and S. Belongie. *Integral Channel Features*. In British Machine Vision Conference (BMVC'09), London,UK, September 2009.

[Dollár 2012] P. Dollár, C. Wojek, B. Schiele and P. Perona. *Pedestrian Detection: An Evaluation of the State of the Art*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pages 743–761, 2012.

[Duval 2013] L. F. Duval. *Hierarchical Cooperation between Fixed View and PTZ Camers*. Rapport laas, LAAS, July 2013.

[Ekinci 2003] M. Ekinci and E. Gedikli. *Background Estimation Based People Detection and Tracking for Video Surveillance*. In A. Yazıcı and C. Şener, editeurs, Computer and Information Sciences - ISCIS 2003, volume 2869 of *Lecture Notes in Computer Science*, pages 421–429. Springer Berlin Heidelberg, 2003.

[Elfring 2013] J. Elfring, S. van den Dries, M. J. G. van de Molengraft and M. Steinbuch. *Semantic World Modeling using Probabilistic Multiple Hypothesis Anchoring*. Robotics and Autonomous Systems, vol. 61, no. 2, pages 95–105, 2013.

[Enzweiler 2009] M. Enzweiler and D.M. Gavrila. *Monocular Pedestrian Detection: Survey and Experiments*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, pages 2179–2195, 2009.

[Ess 2010] A. Ess, K. Schindler, B. Leibe and L. Van Gool. *Object Detection and Tracking for Autonomous Navigation in Dynamic Environments*. The International Journal of Robotics Research, vol. 29, no. 14, pages 1707–1725, 2010.

[Everett 2003] H.R. Everett. *Robotic Security Systems*. IEEE Instrumentation Measurement Magazine, vol. 6, no. 4, pages 30–34, 2003.

[Everingham 2010] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The Pascal Visual Object Classes (VOC) Challenge*. International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, 2010.

[Felzenszwalb 2005] P. F. Felzenszwalb and D. P. Huttenlocher. *Pictorial Structures for Object Recognition*. International Journal of Computer Vision, vol. 61, no. 1, pages 55–79, 2005.

[Felzenszwalb 2010a] P. F. Felzenszwalb, R. B. Girshick and D. McAllester. *Cascade Object Detection with Deformable Part Models*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), San Francisco, CA, USA, June 2010.

[Felzenszwalb 2010b] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan. *Object Detection with Discriminatively Trained Part-Based Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pages 1627–1645, 2010.

[Fischler 1973] M. A. Fischler and R. A. Elschlager. *The Representation and Matching of Pictorial Structures*. IEEE Transactions on Computers, vol. C-22, no. 1, pages 67–92, 1973.

[Fisher 1936] R. A. Fisher. *The Use of Multiple Measurements in Taxonomic Problems*. Annals of Eugenics, vol. 7, no. 7, pages 179–188, 1936.

[Foucher 2011] S. Foucher, M. Lalonde and L. Gagnon. *A System for Airport Surveillance: Detection of People Running, Abandoned Objects, and Pointing Gestures*. In Proceedings of SPIE, volume 8056, June 2011.

[Fritsch 2003] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink and G. Sagerer. *Multi-modal anchoring for human-robot interaction*. Robotics and Autonomous Systems, vol. 43, no. 2-3, pages 133–147, 2003.

[Fritsch 2004] J. Fritsch, M. Kleinehagenbrock, S. Lang, G. A. Fink and G. Sagerer. *Audio-visual Person Tracking with a Mobile Robot*. In International Conference on Intelligent Autonomous Systems (IAS'04), Amsterdam, The Netherlands, March 2004.

[Gabriel 2003] P. F. Gabriel, J. G. Verly, J. H. Piater and A. Genon. *The State of the Art in Multiple Object Tracking under Occlusion in Video Sequences*. In Advanced Concepts for Intelligent Vision Systems (ACIVS'03), Ghent, Belgium, September 2003.

[Gavrila 1999] D.M. Gavrila and V. Philomin. *Real-time Object Detection for "Smart" Vehicles*. In IEEE International Conference on Computer Vision (ICCV'99), volume 1, Corfu, Greece, September 1999.

[Gavrila 2000] D.M. Gavrila. *Pedestrian Detection from a Moving Vehicle*. In European Conference on Computer Vision (ECCV'00), Dublin, Ireland, June 2000.

[Germa 2010] T. Germa, F. Lerasle, N. Ouadah and V. Cadenat. *Vision and RFID data fusion for tracking people in crowds by a mobile robot.* Computer Vision and Image Understanding, vol. 114, no. 6, pages 641–651, 2010.

[Gerónimo 2007] D. Gerónimo, A. M. López, D. Ponsa and A. D. Sappa. *Haar Wavelets and Edge Orientation Histograms for On-Board Pedestrian Detection.* In Iberian Conference Pattern Recognition and Image Analysis (IbPRIA'07), Girona, Spain, June 2007.

[Gerónimo 2010a] D. Gerónimo, A.M. López, A.D. Sappa and T. Graf. *Survey of Pedestrian Detection for Advanced Driver Assistance Systems.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 7, pages 1239–1258, 2010.

[Gerónimo 2010b] D. Gerónimo, A. D. Sappa, D. Ponsa and A. M. López. *2D-3D-based on-board Pedestrian Detection System.* Computer Vision and Image Understanding, vol. 114, no. 5, pages 583–595, May 2010.

[Gerónimo 2012] D. Gerónimo, F. Lerasle and A. López. *State-Driven Particle Filter for Multi-person Tracking.* In Advanced Concepts for Intelligent Vision Systems (ACIVS'12), Brno, Czech Republic, September 2012.

[Gockley 2007] R. Gockley, J. Forlizzi and R. Simmons. *Natural person-following behavior for social robots.* In International Conference on Human-Robot Interaction (HRI'07), Arlington, VA, USA, 2007.

[Gross 2009] H.-M. Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten and A. Bley. *TOOMAS: Interactive Shopping Guide Robots in Everyday Use - Final Implementation and Experiences from Long-term Field Trials.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09), St. Louis, MO, USA, October 2009.

[Gurobi 2013] Gurobi Optimization, Inc. *Gurobi Optimizer Reference Manual*, 2013. http://www.gurobi.com.

[Haga 2004] T. Haga, K. Sumi and Y. Yagi. *Human Detection in Outdoor Scene using Spatio-temporal Motion Analysis.* In International Conference on Pattern Recognition (ICPR'04), volume 4, Cambridge, UK, August 2004.

[Hahnel 2004] D. Hahnel, W. Burgard, D. Fox, K. Fishkin and M. Philipose. *Mapping and Localization with RFID Technology.* In IEEE International Conference on Robotics and Automation (ICRA'04), volume 1, Barcelona, Spain, April 2004.

[Hastings 1970] W. K. Hastings. *Monte Carlo Sampling Methods using Markov Chains and their Applications.* Biometrika, vol. 57, no. 1, pages 97–109, 1970.

[Hayduk 1978] L. A. Hayduk. *Personal Space: An Evaluative and Orienting Overview.* Psychological Bulletin, vol. 85, no. 1, pages 117–134, 1978.

[Heikkilä 2009] M. Heikkilä, M. Pietikäinen and C. Schmid. *Description of Interest rRegions with Local Binary Patterns.* Pattern Recognition, vol. 42, no. 3, pages 425 – 436, 2009.

[Hu 2004] W. Hu, T. Tan, L. Wang and S. Maybank. *A Survey on Visual Surveillance of Object Motion and Behaviors.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 34, no. 3, pages 334 –352, August 2004.

[Huang 2008a] C. Huang, B. Wu and R. Nevatia. *Robust Object Tracking by Hierarchical Association of Detection Responses*. In European Conference on Computer Vision (ECCV'08), Marseille, France, October 2008.

[Huang 2008b] Y. Huang, J. Benesty and J. Chen. *Time Delay Estimation and Source Localization*. In J. Benesty, M. M. Sondhi and Y. Huang, editeurs, Springer Handbook of Speech Processing, pages 1043–1063. Springer Berlin Heidelberg, 2008.

[Hussain 2010] S. Hussain and B. Triggs. *Feature Sets and Dimensionality Reduction for Visual Object Detection*. In British Machine Vision Conference (BMVC'10), Aberystwyth, UK, August 2010.

[Ikemura 2011] S. Ikemura and H. Fujiyoshi. *Real-Time Human Detection Using Relational Depth Similarity Features*. In Asian Conference on Computer Vision (ACCV'11), Queensland, New Zealand, November 2011.

[Isard 2001] M. Isard and J. MacCormick. *BraMBLe: a Bayesian Multiple-blob Tracker*. In International Conference on Computer Vision (ICCV'01), volume 2, Vancouver, Canada, July 2001.

[Jarvis 2008] R. Jarvis. *Robotic Inspection and Safe Removal of Suspicious/Abandoned Luggage*. In IEEE Conference on Robotics, Automation and Mechatronics (RAM'08), 2008.

[Jayawardena 2010] C. Jayawardena, I. H. Kuo, U. Unger, A. Igic, R. Wong, C.I. Watson, R.Q. Stafford, E. Broadbent, P. Tiwari, J. Warren, J. Sohn and B.A. MacDonald. *Deployment of a Service Robot to Help Older People*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10), Taipei, Taiwan, October 2010.

[Joachims 1999] T. Joachims. *Making Large-scale Support Vector Machine Learning Practical*. In B. Schölkopf, C. J. Burges and A. J. Smola, editeurs, Advances in Kernel Methods, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[Jourdheuil 2012] L. Jourdheuil, N. Allezard, T. Chateau and T. Chesnais. *Heterogeneous AdaBoost with Real-time Constraints - Application to the Detection of Pedestrians by Stereovision*. In International Conference on Computer Vision Theory and Applications (VISAPP'12), Rome, Italy, February 2012.

[Kamijo 2000] S. Kamijo, Y. Matsushita, K. Ikeuchi and M. Sakauchi. *Traffic Monitoring and Accident Detection at Intersections*. IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 2, pages 108–118, 2000.

[Kanda 2010] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro and N. Hagita. *A Communication Robot in a Shopping Mall*. IEEE Transactions on Robotics, vol. 26, no. 5, pages 897–913, oct. 2010.

[Khan 2003] Z. Khan, T. R. Balch and F. Dellaert. *Efficient Particle Filter based Tracking of Multiple Interacting Targets using an MRF-based Motion Model*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03), Las Vegas, NV, USA, October 2003.

[Khan 2005] Z. Khan, T. Balch and T. Dellaert. *MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pages 1805–1918, 2005.

[Kobilarov 2006] M. Kobilarov, G. Sukhatme, J. Hyans and P. Bataria. *People Tracking and Following with a Mobile Robot using an Omnidirectional Camera and a Laser.* In IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, Florida, USA, May 2006.

[Kuhn 1955] H. W. Kuhn. *The Hungarian Method for the Assignment Problem.* Naval Research Logistics Quarterly, vol. 2, no. 1-2, pages 83–97, 1955.

[Lee 2006] J. Lee, T. Tsubouchi, K. Yamamoto and S. Egawa. *People Tracking Using a Robot in Motion with Laser Range Finder.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06), Beijing, China, October 2006.

[Lefebvre 2004] T. Lefebvre, H. Bruyninckx and J. De Schutter. *Kalman Filters for Non-Linear Systems: a Comparison of Performance.* International Journal of Control, vol. 77, no. 7, pages 639–653, 2004.

[Leibe 2007] B. Leibe, K. Schindler and L. Van Gool. *Coupled Detection and Trajectory Estimation for Multi-Object Tracking.* In IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, October 2007.

[Leibe 2008] B. Leibe, A. Leonardis and B. Schiele. *Robust Object Detection with Interleaved Categorization and Segmentation.* International Journal of Computer Vision, vol. 77, no. 1-3, pages 259–289, May 2008.

[Levi 2004] K. Levi and Yair Weiss. *Learning Object Detection from a Small Number of Examples: The Importance of Good Features.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04), Washington, DC, USA, June 2004.

[Li 2008] Y. Y. Li and L.E. Parker. *Detecting and Monitoring Time-related Abnormal Events using a Wireless Sensor Network and Mobile Robot.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08), Nice, France, September 2008.

[Lienhart 2002] R. Lienhart and J. Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection.* In IEEE International Conference on Image Processing (ICIP'02), New York, USA, September 2002.

[Lin 2008] Z. Lin and L. S. Davis. *A Pose-Invariant Descriptor for Human Detection and Segmentation.* In European Conference on Computer Vision (ECCV'08), Marseille, France, October 2008.

[Luber 2010] M. Luber, J.A. Stork, G.D. Tipaldi and K.O. Arras. *People Tracking with Human Motion Predictions from Social Forces.* In International Conference on Robotics and Automation (ICRA'10), Anchorage, AK, USA, May 2010.

[Luber 2011] M. Luber, L. Spinello and K. O. Arras. *People tracking in RGB-D Data with online boosted target models.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11), San Francisco, CA, USA, September 2011.

[Madrigal 2013] F. Madrigal and J.-B. Hayet. *Evaluation of Multiple Motion Models for Multiple Pedestrian Visual Tracking.* In IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'13), Krakow, Poland, August 2013.

[Maji 2008] S. Maji, A.C. Berg and J. Malik. *Classification using Intersection Kernel Support Vector Machines is Efficient.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), Anchorage, AL, USA, June 2008.

[Manzanera 2007] A. Manzanera. *Sigma - Delta background subtraction and the Zipf law.* In Iberoamerican Congress on Pattern Recognition (CIARP'07), Valparaiso, Chile, November 2007.

[Marikhu 2013] R. Marikhu, J. Moonrinta, M. Ekpanyapong, M. Dailey and S. Siddhichai. *Police Eyes: Real World Automated Detection of Traffic Violations.* In International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON'13), Krabi, Thailand, May 2013.

[Martin 2006] C. Martin, E. Schaffernicht, A. Scheidi and H. Gross. *Multi-modal Sensor Fusion using a Probabilistic Aggregation Scheme for People Detection and Tracking.* Robotics and Autonomous Systems, vol. 54, no. 9, pages 721–728, 2006.

[Meden 2012] B. Meden, F. Lerasle and P. Sayd. *MCMC Supervision for People Re-identification in Nonoverlapping Cameras.* In British Machine Vision Conference (BMVC'12), Surrey, UK, September 2012. BMVA Press.

[Mekonnen 2011] A. A. Mekonnen, F. Lerasle and I. Zuriarrain. *Multi-modal Person Detection and Tracking from a Mobile Robot in Crowded Environment.* In International Conference on Computer Vision Theory and Applications (VISAPP'11), Algarve, Portugal, March 2011.

[Mekonnen 2012] A. A. Mekonnen, F Lerasle and A Herbulot. *Coopération entre un robot mobile et des caméras d'ambiance pour le suivi multi-personnes.* In congrès francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'12), Lyon, France, Janvier 2012.

[Mekonnen 2013a] A. A. Mekonnen, C. Briand, F. Lerasle and A. Herbulot. *Fast HOG based Person Detection devoted to a Mobile Robot with a Spherical Camera.* In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'13), Tokyo,Japan, November 2013.

[Mekonnen 2013b] A. A. Mekonnen, A. Herbulot and F. Lerasle. *Coopération entre perception déportée et embarquée sur un robot guide pour l'aide à sa navigation.* Revue d'Intelligence Artificielle, vol. 27, no. 1, pages 65–93, 2013.

[Mekonnen 2013c] A. A. Mekonnen, F Lerasle and A. Herbulot. *Cooperative Passers-by Tracking with a Mobile Robot and External Cameras.* Computer Vision and Image Understanding, vol. 117, no. 10, pages 1229–1244, 2013.

[Mekonnen 2013d] A. A. Mekonnen, F. Lerasle and A. Herbulot. *External Cameras and a Mobile Robot for Enhanced Multi-person Tracking.* In International Conference on Computer Vision Theory and Applications (VISAPP'13), Barcelona, Spain, February 2013.

[Mekonnen 2013e] A. A. Mekonnen, F. Lerasle and A. Herbulot. *Pareto-Front Analysis and AdaBoost for Person Detection using Heterogenous Features.* In IEEE International Conference on Systems, Man, and Cybernetics (SMC'13), Manchester, UK, October 2013.

[Mekonnen 2013f] A. A. Mekonnen, F. Lerasle and A. Herbulot. *Person Detection with a Computation Time Weighted AdaBoost.* In Advanced Concepts in Intelligent Vision Systems (ACIVS'13), Poznan,Poland, October 2013.

[Mekonnen 2014a] A. A. Mekonnen, F. Lerasle and A. Herbulot. *People Detection with Heterogeneous Features and Explicit Optimization on Computation Time*. In International Conference on Pattern Recognition (ICPR'14), Stockholm, Sweden, August 2014.

[Mekonnen 2014b] A. A. Mekonnen, F. Lerasle, A. Herbulot and C. Briand. *Détection de personnes par apprentissage de descripteurs hétérogènes sous des considérations CPU*. In congrès francophone sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'14), Rouen, France, Juillet 2014.

[Micheloni 2003] C. Micheloni, G.L. Foresti and L. Snidaro. *A Cooperative Multicamera System for Video-surveillance of Parking Lots*. In IEE Symposium on Intelligence Distributed Surveillance Systems, London, UK, February 2003.

[Microsoft 2010] Microsoft Corporation Redmond WA. *Kinect for XBOX*, 2010. http://www.xbox.com/en-US/Kinect, retrieved: March 02, 2012.

[Mikolajczyk 2004] Krystian Mikolajczyk, Cordelia Schmid and Andrew Zisserman. *Human Detection based on a Probabilistic Assembly of Robust Part Detectors*. In European Conference on Computer Vision (ECCV'04), volume I, Prague, Czech Republic, May 2004.

[Milch 2001] S. Milch and M. Hehrens. *Pedestrian Detection with Radar and Computer Vision*. PAL 2001-Progress in Automobile Lighting, vol. 9, 2001.

[Minguez 2004] J. Minguez and L. Montano. *Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios*. IEEE Transactions on Robotics and Automation, vol. 20, no. 1, pages 45 – 59, February, 2004.

[Mitzel 2010] D. Mitzel, E. Horbert, A. Ess and B. Leibe. *Multi-person Tracking with Sparse Detection and Continuous Segmentation*. In European Conference on Computer Vision (ECCV'10), Crete, Greece, September 2010.

[Mogelmose 2012] A. Mogelmose, A. Prioletti, M.M. Trivedi, A. Broggi and T.B. Moeslund. *Two-stage Part-based Pedestrian Detection*. In IEEE International Conference on Intelligent Transportation Systems (ITSC'12), Anchorage, AK, USA, September 2012.

[Mohan 2001] A. Mohan, C. Papageorgiou and T. Poggio. *Example-Based Object Detection in Images by Components*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 4, pages 349–361, April 2001.

[MSDN] Microsoft Developer Network. *Kinect for Windows Sensor Components and Specifications*. http://msdn.microsoft.com/en-us/library/jj131033.aspx, retrieved: Novemeber 02, 2013.

[Mu 2008] Y. Mu, S. Yan, Y. Liu, T. Huang and B. Zhou. *Discriminative Local Binary Patterns for Human Detection in Personal Album*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), Anchorage, AK, USA, June 2008.

[Muñoz-Salinas 2007] R. Muñoz-Salinas, E. Aguirre and M. García-Silvente. *People Detection and Tracking using Stereo Vision and Color*. Image and Vision Computing, vol. 25, no. 6, pages 995 – 1007, 2007.

[Nakadai 2001] Kazuhiro Nakadai, Ken-ichi Hidai, Hiroshi Mizoguchi, Hiroshi G. Okuno and Hiroaki Kitano. *Real-time Auditory and Visual Multiple-object Tracking for Humanoids*. In International Joint Conference on Artificial Intelligence (IJCAI'01), Seattle, WA, USA, August 2001.

[Ojala 1996] T. Ojala, M. Pietikäinen and D. Harwood. *A Comparative Study of Texture Measures with Classification based on Featured Distributions*. Pattern Recognition, vol. 29, no. 1, pages 51 – 59, 1996.

[OpenCv] OpenCv Developers Team. *OpenCv: Open Source Computer Vision Library*. http://www.opencv.org.

[Paillet 2013] P. Paillet, R. Audigier, F. Lerasle and Q. C. Pham. *IMM-Based Tracking and Latency Control with Off-the-Shelf IP PTZ Camera*. In Advanced Concepts in Intelligent Vision Systems (ACIVS'13), Poznan,Poland, October 2013.

[Paisitkriangkrai 2008] S. Paisitkriangkrai, Chunhua Shen and Jian Zhang. *Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 8, pages 1140–1151, 2008.

[Paleček 2012] K. Paleček, D. Gerónimo and F. Lerasle. *Pre-attention Cues for Person Detection*. In A. Esposito, A. M.. Esposito, A. Vinciarelli, R. Hoffmann and V. C. Müller, editeurs, Cognitive Behavioural Systems, volume 7403 of *Lecture Notes in Computer Science*, pages 225–235. Springer Berlin Heidelberg, 2012.

[Pan 2013] H. Pan, Y. Zhu and L. Xia. *Efficient and Accurate Face Detection using Heterogeneous Feature Descriptors and Feature Selection*. Computer Vision and Image Understanding, vol. 117, no. 1, pages 12 – 28, 2013.

[Pandey 2010] A. K. Pandey and R. Alami. *A Framework Towards a Socially Aware Mobile Robot Motion in Human-Centered Dynamic Environment*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10), Taipei, Taiwan, October 2010.

[Papageorgiou 2000] C. Papageorgiou and T. Poggio. *A Trainable System for Object Detection*. IJCV, vol. 38, no. 1, pages 15–33, 2000.

[Paragios 2000] N. Paragios and R. Deriche. *Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 3, pages 266–280, March 2000.

[Park 2010] D. Park, D. Ramanan and C. Fowlkes. *Multiresolution Models for Object Detection*. In European Conference on Computer Vision (ECCV'10), Crete, Greece, September 2010.

[Pérez 2002] P. Pérez, C. Hue, J. Vermaak and M. Gangnet. *Color-Based Probabilistic Tracking*. In European Conference on Computer Vision (ECCV'02), Copenhagen, Denmark, May 2002.

[Perez 2004] P. Perez, J. Vermaak and A. Blake. *Data Dusion for Visual Tracking with Particles*. Proceedings of the IEEE, vol. 92, no. 3, pages 495–513, 2004.

[Piccardi 2004] M. Piccardi. *Background Subtraction Techniques: A Review*. In IEEE International Conference on Systems, Man and Cybernetics (SMC'04), volume 4, The Hague, The Netherlands, October 2004.

[Point Grey Inc. 2012] Point Grey Inc. *Ladybug2*. http://www.ptgrey.com/products/ladybug2/, 2012. [Online; accessed 29-January-2013].

[Porikli 2008] Fatih Porikli, Yuri Ivanov and Tetsuji Haga. *Robust Abandoned Object Detection Using Dual Foregrounds*. EURASIP Journal on Advances in Signal Processing, vol. 2008, January 2008.

[Pozzobon 1999] A. Pozzobon, G. Sciutto and V. Recagno. *Security in Ports: the User Requirements for Surveillance System.* In C. Regazzoni, G. Fabri and G. Vernazza, editeurs, Advanced Video-Based Surveillance Systems, volume 488 of *The Springer International Series in Engineering and Computer Science*, pages 18–26. Springer US, 1999.

[Rajasekaran 2010] M. P. Rajasekaran, S. Radhakrishnan and P. Subbaraj. *Sensor Grid Applications in Patient Monitoring.* Future Generation Computer Systems, vol. 26, no. 4, pages 569–575, April 2010.

[Rasmussen 2001] C. Rasmussen and G. D. Hager. *Probabilistic Data Association Methods for Tracking Complex Visual Objects.* IEEE Transactions in Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pages 560–576, 2001.

[Räty 2010] T.D. Räty. *Survey on Contemporary Remote Surveillance Systems for Public Safety.* IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, no. 5, pages 493–515, 2010.

[Reid 1979] D. Reid. *An Algorithm for Tracking Multiple Targets.* IEEE Transactions on Automatic Control, vol. 24, no. 6, pages 843–854, December 1979.

[Roboporter] Yasakawa Electric. *Roboporter.* http://robot.watch.impress.co.jp/cda/news/2008/02/13/900.html, retrieved: December 20, 2013.

[Ronetti 2000] N. Ronetti and C. Dambra. *Railway Station Surveillance: The Italian Case.* In G. L. Foresti, P. Mähönen and C. Regazzoni, editeurs, Multimedia Video-Based Surveillance Systems, volume 573 of *The Springer International Series in Engineering and Computer Science*, pages 13–20. Springer US, 2000.

[Roy 1986] T. J. Van Roy and L. A. Wolsey. *Valid Inequalities for Mixed 0-1 Programs.* Discrete Applied Mathematics, vol. 14, no. 7, pages 199–213, 1986.

[Sabzmeydani 2007] P. Sabzmeydani and G. Mori. *Detecting Pedestrians by Learning Shapelet Features.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07), Minneapolis, MN, USA, June 2007.

[Salas 2011] J. Salas and C. Tomasi. *People Detection Using Color and Depth Images.* In J. Martínez-Trinidad, J. Carrasco-Ochoa, C. Ben-Youssef Brants and E. R. Hancock, editeurs, Pattern Recognition, volume 6718 of *Lecture Notes in Computer Science*, pages 127–135. Springer Berlin Heidelberg, 2011.

[Salvi 2010] J. Salvi, S. Fernandez, T. Pribanic and X. Llado. *A State of the Art in Structured Light Patterns for Surface Profilometry.* Pattern Recognition, vol. 43, no. 8, pages 2666 – 2680, 2010.

[San Jose del Cabo Airport] San Jose del Cabo Airport. *San Jose del Cabo Airport Commercial Area.* http://sanzpont.wordpress.com/2013/07/04/commercial-area-san-jose-del-cabo-airport/, retrieved: December 20, 2013.

[Satpathy 2013] A. Satpathy, X. Jiang and H. L. Eng. *Human Detection using Discriminative and Robust Local Binary Pattern.* In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'13), Vancouver, Canada, May 2013.

[Schapire 2003] R. E. Schapire. *The Boosting Approach to Machine Learning: An Overview.* Lecture Notes in Statistics, pages 149–172, 2003.

[Schiele 2009] B. Schiele, M. Andriluka, N. Majer, S. Roth and C. Wojek. *Visual People Detection: Different Models, Comparison and Discussion*. In IEEE ICRA Workshop on People Detection and Tracking, Kobe, Japan, May 2009.

[Scholkopf 2001] B. Scholkopf and A. J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA, 2001.

[Schwartz 2009] W. R. Schwartz, A. Kembhavi, D. Harwood and L. S. Davis. *Human Detection using Partial Least Squares Analysis*. In IEEE International Conference on Computer Vision (ICCV'09), Kyoto, Japan, October 2009.

[Scotti 2005] G. Scotti, L. Marcenaro, C. Coelho, F. Selvaggi and C.S. Regazzoni. *Dual Camera Intelligent Sensor for High Definition 360 Degrees Surveillance*. IEE Proceedings - Vision, Image and Signal Processing, vol. 152, no. 2, pages 250–257, 2005.

[Sisbot 2007] E. A. Sisbot, L. F. Marin-Urias, R. Alami and T. Simeon. *A Human Aware Mobile Robot Motion Planner*. IEEE Transactions on Robotics, vol. 23, no. 5, pages 874–883, 2007.

[Smith 2005] K. Smith, D. Gatica-Perez and J. M. Odobez. *Using Particles to Track Varying Numbers of Interacting People*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, June 2005.

[Smith 2006] D. Smith and S. Singh. *Approaches to Multisensor Data Fusion in Target Tracking: A Survey*. IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 12, pages 1696–1710, 2006.

[Smith 2007] K. C. Smith. *Bayesian Methods for Visual Multi-object Tracking with Applications to Human Activity Recognition*. PhD thesis, Infoscience|Ecole Polytechnique Federale de Lausanne (Switzerland), 2007.

[Spinello 2011] Luciano Spinello and Kai O. Arras. *People detection in RGB-D data*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11), San Francisco, CA, USA, September 2011.

[Szarvas 2005] M. Szarvas, A. Yoshizawa, M. Yamamoto and J. Ogata. *Pedestrian Detection with Convolutional Neural Networks*. In IEEE Intelligent Vehicles Symposium (IV'05), Las Vegas, NV, USA, June 2005.

[T-34] Tmsuk and Alacom. *T-34 Security Robot*. http://www.dailymail.co.uk/sciencetech/article-1126605/Security-robot-nets-burglars-spider-web-spray.html, retrieved: December 20, 2013.

[Tanaka 2004] K. Tanaka and E. Kondo. *Vision-based Multi-Person Tracking by using MCMC-PF and RRF in Office Environments*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04), volume 1, Sendai, Japan, September 2004.

[Tang 2012] D. Tang, Y. Liu and T. Kim. *Fast Pedestrian Detection by Cascaded Random Forest with Dominant Orientation Templates*. In British Machine Vision Conference (BMVC'09), London, UK, September 2012.

[Teixeira 2010] T. Teixeira, G. Dublon and A. Savvides. *A Survey of Human-sensing: Methods for Detecting Presence, Count, Location, Track, and Identity*. Technical report, ENALAB, Yale University, 2010.

[Toth 2003] D. Toth and T. Aach. *Detection and Recognition of Moving Objects using Statistical Motion Detection and Fourier Descriptors*. In International Conference on Image Analysis and Processing (ICIAP'03), Mantova, Italy, September 2003.

[Treptow 2005] A. Treptow, G. Cielniak and T. Duckett. *Active People Recognition using Thermal and Grey Images on a Mobile Security Robot*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05), Edmonton, Canada, August 2005.

[Treptow 2006] A. Treptow, G. Cielniak and T. Duckett. *Real-time People Tracking for Mobile Robots using Thermal Vision*. Robotics and Autonomous Systems, vol. 54, no. 9, pages 729–739, 2006.

[Tseng 2002] B. L. Tseng, C.-Y. Lin and J. R. Smith. *Real-time Video Surveillance for Traffic Monitoring using Virtual Line Analysis*. In IEEE International Conference on Multimedia and Expo (ICME'02), volume 2, Lausanne, Switzerland, August 2002.

[Tuzel 2008] O. Tuzel, F. Porikli and P. Meer. *Pedestrian Detection via Classification on Riemannian Manifolds*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 10, pages 1713–1727, 2008.

[UrRehman 2012] A. UrRehman, S.M. Naqvi, R. Phan, W. Wang and J. Chambers. *MCMC-PF Based Multiple Head Tracking in a Room Environment*. In BMVC Computer Vision Workshop (BMVW'12), Guildford, UK, September 2012.

[Valera 2005] M. Valera and S.A. Velastin. *Intelligent Distributed Surveillance Systems: A Review*. IEE Proceedings - Vision, Image and Signal Processing, vol. 152, no. 2, pages 192 – 204, april 2005.

[Vapnik 1995] Vladimir N. Vapnik. The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Vasquez 2012] D. Vasquez, P. Stein, J. Rios-Martinez, A. Escobedo, A. Spalanzani and C. Laugier. *Human Aware Navigation for Assistive Robotics*. In International Symposium on Experimental Robotics (ISER'12), Québec, Canada, June 2012.

[Viola 2004] P. A. Viola and M. J. Jones. *Robust Real-Time Face Detection*. International Journal of Computer Vision, vol. 57, no. 2, pages 137–154, 2004.

[Viola 2005] P. Viola, M. J. Jones and D. Snow. *Detecting Pedestrians Using Patterns of Motion and Appearance*. International Journal of Computer Vision, vol. 63, no. 2, pages 153–161, 2005.

[Volkhardt 2013] M Volkhardt, C. Weinrich and H.-M. Gross. *Multi-modal People Tracking on a Mobile Companion Robot*. In European Conference on Mobile Robots (ECMR'13), Barcelona, Spain, September 2013.

[Walk 2010] S. Walk, N. Majer, K. Schindler and B. Schiele. *New Features and Insights for Pedestrian Detection*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10), San Francisco, CA, USA, June 2010.

[Wang 2009] X. Wang, T.X. Han and S. Yan. *An HOG-LBP human detector with partial occlusion handling*. In IEEE International Conference on Computer Vision (ICCV'09), Kyoto, Japan, October 2009.

[Wang 2013] X. Wang. *Intelligent Multi-camera Video Surveillance: A Review.* Pattern Recognition Letters, vol. 34, no. 1, pages 3 – 19, 2013.

[Wojek 2008] C. Wojek and B. Schiele. *A Performance Evaluation of Single and Multi-feature People Detection.* In DAGM-Symposium, Munich, Germany, June 2008.

[Wolsey 2003] L. A. Wolsey. *Strong Formulations for Mixed Integer Programs: Valid Inequalities and Extended Formulations.* Mathematical Programming, vol. 97, no. 7, pages 423–447, 2003.

[Wu 2005] B. Wu and R. Nevatia. *Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors.* In IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, October 2005.

[Wu 2007] B. Wu and R. Nevatia. *Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors.* International Journal of Computer Vision, vol. 75, no. 2, pages 247–266, 2007.

[Wu 2008] B. Wu and R. Nevatia. *Optimizing Discrimination-Efficiency Tradeoff in Integrating Heterogeneous Local Features for Object Detection.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), Anchorage, AK, USA, June 2008.

[Wu 2009] X. Wu, H. Gong, P. Chen, Z. Zhi and Y. Xu. *Intelligent Household Surveillance Robot.* In IEEE International Conference on Robotics and Biomimetics (ROBIO'09), Guilin, Guangxi, China, December 2009.

[Xavier 2005] J. Xavier, M. Pacheco, D. Castro and A. Ruano. *Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver.* In IEEE International Conference on Robotics and Automation (ICRA'05), Barcelona, Spain, 2005.

[Yamada 2005] N. Yamada, Y. Tanaka and K. Nishikawa. *Radar Cross Section for Pedestrian in 76GHz Band.* In European Microwave Conference (EuMC'05), Paris, France, October 2005.

[Yao 2009] J. Yao and J.-M. Odobez. *Multi-Person Bayesian Tracking with Multiple Cameras.* In H. Aghajan and A. Cavallaro, editeurs, Multi-Camera Networks, pages 363 – 388. Academic Press, Oxford, 2009.

[Yasushi 1999] YAGI Yasushi. *Omnidirectional Sensing and Its Applications.* IEICE Transactions on Information and Systems, vol. 82, no. 3, pages 568–579, 1999.

[Zajdel 2005] W. Zajdel, Z. Zivkovic and B. Kröse. *Keeping Track of Humans: Have I Seen This Person Before?* In IEEE International Conference in Robotics and Automation (ICRA'05), Barcelona, Spain, July 2005.

[Zambanini 2009] S. Zambanini, P. Blauensteiner and M. Kampel. *Automated Multi-camera Surveillance for the Prevention and Investigation of Bank Robberies in Austria: A Case Study.* In International Conference on Crime Detection and Prevention (ICDP'09), London, UK, December 2009.

[Zetik 2006] R. Zetik, S. Crabbe, J. Krajnak, P. Peyerl, J. Sachs and R. Thomä. *Detection and Localization of Persons behind Obstacles using M-sequence Through-the-Wall Radar.* In Proceedings of SPIE, volume 6201, May 2006.

[Zhang 2012] J. Zhang, L. L. Presti and S. Sclaroff. *Online Multi-person Tracking by Tracker Hierarchy*. In IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'12), Beijing, China, 2012 2012.

[Zhao 1999] L. Zhao and C. Thorpe. *Stereo-and Neural Network-based Pedestrian Detection*. In IEEE International Conference on Intelligent Transportation Systems (ITSC'99), Tokyo, Japan, October 1999.

[Zhao 2007] G. Zhao and M. Pietikainen. *Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pages 915–928, 2007.

[Zhu 2006] Q. Zhu, M.-C. Yeh, K.-T. Cheng and S. Avidan. *Fast Human Detection Using a Cascade of Histograms of Oriented Gradients*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, June 2006.

[Zivkovic 2007] Z. Zivkovic and B. Kröse. *Part Based People Detection using 2D Range Data and Images*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07), San Diego, CA, USA, November 2007.

[Zivkovic 2008] Z. Zivkovic and B. Kröse. *People Detection Using Multiple Sensors on a Mobile Robot*. In Danica Kragic and Ville Kyrki, editeurs, Unifying Perspectives in Computational and Robot Vision, volume 8, pages 25–39. Springer US, 2008.

[Zouba 2009] N. Zouba, F. Bremond and M. Thonnat. *Multisensor Fusion for Monitoring Elderly Activities at Home*. In IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'09), Genova, Italy, September 2009.

[Zuriarrain 2013] I. Zuriarrain, A. A. Mekonnen, F. Lerasle and N. Arana. *Tracking-by-Detection of Multiple Persons by a Resample-Move Particle Filter*. Machine Vision and Applications, vol. 24, no. 8, pages 1751–1765, 2013.

Titre :
Coopération de réseaux de caméras ambiantes et de vision embarquée sur robot mobile pour la surveillance de lieux publics

Directeurs de Thèse :
Ariane HERBULOT, Maître de Conférences, Université Toulouse III - Paul Sabatier
Frédéric LERASLE, Professeur des Universités, Université Toulouse III - Paul Sabatier

Lieu et Date de Soutenance :
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, le 11 Mars 2014

Résumé :

Actuellement, il y a une demande croissante pour le déploiement de robots mobile dans des lieux publics. Pour alimenter cette demande, plusieurs chercheurs ont déployé des systèmes robotiques de prototypes dans des lieux publics comme les hôpitaux, les supermarchés, les musées, et les environnements de bureau. Une principale préoccupation qui ne doit pas être négligé, comme des robots sortent de leur milieu industriel isolé et commencent à interagir avec les humains dans un espace de travail partagé, est une interaction sécuritaire. Pour un robot mobile à avoir un comportement interactif sécuritaire et acceptable - il a besoin de connaître la présence, la localisation et les mouvements de population à mieux comprendre et anticiper leurs intentions et leurs actions. Cette thèse vise à apporter une contribution dans ce sens en mettant l'accent sur les modalités de perception pour détecter et suivre les personnes à proximité d'un robot mobile.

Comme une première contribution, cette thèse présente un système automatisé de détection des personnes visuel optimisé qui prend explicitement la demande de calcul prévue sur le robot en considération. Différentes expériences comparatives sont menées pour mettre clairement en évidence les améliorations de ce détecteur apporte à la table, y compris ses effets sur la réactivité du robot lors de missions en ligne. Dans un deuxiè contribution, la thèse propose et valide un cadre de coopération pour fusionner des informations depuis des caméras ambiant affixé au mur et de capteurs montés sur le robot mobile afin de mieux suivre les personnes dans le voisinage. La même structure est également validée par des données de fusion à partir des différents capteurs sur le robot mobile au cours de l'absence de perception externe. Enfin, nous démontrons les améliorations apportées par les modalités perceptives développés en les déployant sur notre plate-forme robotique et illustrant la capacité du robot à percevoir les gens dans les lieux publics supposés et respecter leur espace personnel pendant la navigation.

Mots-clé :
Systèmes de perception coopérative ; Réseaux de caméras ; Détection de personnes ; Sélection de descripteurs ; Suivi multi-cibles ; Suivi Bayésien ; Apprentissage ; Fusion de données ;

Discipline administrative :
Systèmes embarqués et Robotique

Intitulé et Adresse du Laboratoire :
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) du CNRS
7 Avenue du Colonel Roche, 31400 Toulouse