University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Semantically Integrating Laser and Vision
in Pedestrian Detection

Luciano Rebouças de Oliveira

2010

Semantically Integrating Laser and Vision
in Pedestrian Detection

Luciano Rebouças de Oliveira

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Institute of Systems and Robotics
Department of Electrical and Computer Engineering
University of Coimbra, Portugal

under supervision of

Prof. Dr. Urbano Nunes (advisor)
Prof. Dr. Paulo Peixoto (co-advisor)

To my wife and son

Acknowledgments

In this apparently lonely journey, I owe a lot to several persons my special thanks. When I first came in Coimbra, I had the pleasure to meet Gonçalo Monteiro, who was my lab colleague, during approximately one year. From him I learnt many practical aspects about computer vision and pattern recognition, which helped me during these three years here. I would also like to thank Cristiano Premebida who gave me a very good assistance in getting some documents in Portugal when I still was in Brazil.

During a PhD, the relationship with the advisor is essential to its success. Therefore, I can say Prof. Urbano Nunes' supervision was outstanding, not only because of his friendship but also because he challenged me to reach difficult goals, like NiSIS competition award, for example. He also provided me with all the necessary support to obtain my achievements in this whole period here, always giving me the freedom to follow my own directions. I am also very grateful to Prof. Paulo Peixoto for his technical supervision. I owe a lot to ISR-UC to have provided excelent conditions that allowed me to accomplish the thesis' objectives. This work has been in part supported by the Portuguese Science and Technology Foundation (FCT), under project contract PmITS06 (PTDC/EEA-ACR/72226/2006), and by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), which has supplied me with a scholarship, under process number BEX 4000/05-6.

Many colleagues from ISR-UC have participated indirectly in my work with interesting and fruitful discussions about image processing, pattern recognition or computer vision, such as: Gabriel Pires, Pedro Martins, Hélio Palaio, José Roquette, João Carreira, and Joerg Rett. My office mates, Pedro Miraldo and Abed Malti, and my colleague Rui Caseiro, were of great importance during my last year of thesis, helping me with useful technical advices. I am also very grateful to Marco Silva and Fernando Moita who helped me with the last experiments of this thesis. I can not forget to mention Carlos Belchior and Ricardo Maia and our interesting philosophical discussions. I also thank Cristina and Tito, who supported me on many university issues. Among committed professors at ISR, I would like to say a special thank you to Prof. João Barreto as one who has instigated me to go deeper into scientific problems.

For all these almost 10 years of mental and physical preparation to support daily pressure, I am grateful to my Aikido instructors and colleagues from Brazil. Also, I thank Tim Minaker for useful hints in English writing, and Jacques Chicourel for always encouraging me to come into the business world.

Last but not least, I would like to thank my family, not only for supporting me

thousand of kilometers away from Portugal, but also for being here with me in some crucial moments. I would not be here without my dearly beloved wife, who has given me her dedication, love, comprehension, friendship, complicity and, last year, our son Victor, whose presence has helped me to keep my mind focused.

Abstract

Perception systems are now a reality with the deployment of camera-based object detection and lane departure systems in top-of-the-line vehicles. Although these systems are intended to aid the driver in hazardous situations, much has yet to be done in order to make them completely reliable in several circumstances. In this regard, multi-sensor architectures may bring complementarity and redundancy, making perception systems more robust. This way, the aim of this thesis is to contribute with a novel perception system based on laser scanner and vision for object detection applied to urban scenarios. Although our system can be adapted to recognize any outdoor object, our method's proof of concept is performed on pedestrians, since they represent a challenge because of the variety of their poses, positions, forms, sizes, and colors.

Particularly, fusion of laser and vision in object detection has been accomplished by two main approaches: 1) independent integration of sensor-based features or classifiers; or, 2) after finding a region of interest (ROI) with the laser, an image classifier is used to name the projected ROI as object or non-object. Both methods rely on independent and identically distributed assumptions, which can be unrealistic in many situations. In this thesis, we propose a fusion approach based on semantic information embodied on many levels. Sensor data fusion is based on spatial relationship of parts-based classifiers integrated with object-centered context, lately modeled via a Markov logic network. The proposed system deals with partial segmentation; it is able to recover depth information even if the laser fails, and the sensor data fusion is modeled through semantic representation – characteristics not found on existing approaches. The performance of the proposed method is assessed by receiver operating characteristics curves of each component detector, and the semantic fusion method, as well. The results demonstrated the effectiveness of the proposed method over available and gathered data sets in challenging urban scenarios. The use of available data sets were mainly for comparison of our proposed vision ensemble detectors with other methods.

We hope that our work acts as a starting point for others to explore different multi-sensor fusion methods. In addition, the proposed system represents a reliable detection framework wherein tracking systems can be built.

Contents

I Presentation 1

1 Introduction 3

1.1 Motivation 4

1.2 Goals 5

1.3 Key contributions 6

1.4 Chapter map 7

2 Background 9

2.1 Laser-based object detection 9

2.2 Object detection in images 10

2.3 Ensemble of classifiers 12

2.4 Fusion of laser and vision 13

2.5 Data sets 14

2.6 Relation to our work 15

3 Multi-sensor object detection in urban scenes 19

3.1 Proposed framework outline 19

3.2 Experimental setup and collected data sets 20

3.2.1 Gathered data sets 22

3.3 Remarks 23

II The perception system 25

4 Toward improving local detection by ensemble of classifiers 27

4.1 Classifiers 28

4.1.1 MLP and SVM 28

4.1.2 Adaboost 29

4.2 Feature extractors 29

4.2.1 Haar-like 30

4.2.2 HOG 32

4.2.3 LRF . . . 34

4.3 Proposed ensembles . . . 36

4.3.1 HFI . . . 36

4.3.2 HLSM-FINT . . . 37

4.4 Experimental validation . . . 41

4.4.1 Parameter selection of the feature extractors . . . 42

4.4.2 Evaluation of the component classifiers . . . 42

4.4.3 Sensitivity analysis . . . 43

4.4.4 Comparison of classifier fusion methods . . . 45

4.4.5 Evaluation on NiSIS competition . . . 48

4.4.6 Comparison of HLSM-FINT with other methods over Daimler-Chrysler data sets . . . 49

4.4.7 Evaluation of HFI and HLSM-FINT over ISR-UC-imgs data set . . . 50

4.5 Conclusions . . . 53

5 Laser segmentation and labeling: A clustering-based approach 55

5.1 Outline of the laser-based detection system . . . 56

5.2 From laser points to labels . . . 56

5.2.1 Segmenting points . . . 57

5.2.2 Labeling segments . . . 60

5.3 Dealing with occlusion by spatial relationship between laser segments . . . 65

5.4 Conclusions . . . 66

6 Semantic fusion of laser and vision 69

6.1 Sensor-driven detectors . . . 71

6.1.1 Image object detector . . . 72

6.2 Foundations of Markov logic network . . . 76

6.2.1 First-order logic . . . 76

6.2.2 Markov random field . . . 77

6.2.3 Markov logic network . . . 77

6.3 Semantically integrating the parts . . . 80

6.4 Experimental validation . . . 81

6.5 Conclusions . . . 84

III Discussion and conclusions 87

7 Discussion 89

7.1 Effective detection systems in ITS 89

7.2 Toward system implementation 91

7.3 Future directions 91

8 Conclusion 93

IV Appendices 95

A Support vector machines 97

B Classifier fusion methods 103

B.1 Majority vote 103

B.2 Fuzzy integral 103

C Parallel architecture for object detection 105

C.1 Object searching 105

C.1.1 Object classification 107

C.2 Exploring parallelism using Cell 108

C.2.1 Cell architecture 108

C.2.2 Optimizing object detection algorithms 109

References 113

List of Tables

2.1 Some available pedestrian data sets for benchmarking 15

3.1 Characteristics of “ISR-UC-imglidar-sync” data sets 22

3.2 Characteristics of “ISR-UC-imgs” data sets 23

4.1 Exponential error count (E_{ec}) for classifier combinations from the initial ensemble 41

4.2 Comparative results after lighting transformations 46

4.3 Results of the fusion methods over DaimlerChrysler data sets 47

4.4 Result of the fusion methods over transformed DaimlerChrysler data sets . 47

4.5 Summary of the results over DaimlerChrysler and transformed DaimlerChrysler data sets (HR/FAR (%)) 48

4.6 Comparison over DaimlerChrysler data sets 49

5.1 Relabeling rules 61

6.1 First-order formulas used in the semantic fusion. An object “o” can be either an image window or a laser segment. 80

6.2 Summary of the final results 84

List of Figures

1.1 Most of us could identify the two persons under the sunset. But for an image detector, it could be cumbersome to do it, as the persons present either very low resolution in the original image, or they loose their inherent aspect after resized (sided image). 4

2.1 Diversity in ensemble of two classifiers: Image on the top shows the individual classification by two classifiers, while image on the bottom is the result of a classification fusion. It is expected that a fusion method takes the complementarity of the component classifiers into account in a way that the performance of the ensemble is superior to the performance of the individual classifiers. 13

2.2 Some samples of (a) DaimlerChrysler and (b) INRIA data sets. 15

3.1 The proposed framework is composed of 6 main blocks: The image object detection based on our parts-based ensemble detector HLSM-FINT (Chapters 4 and 6); laser segmentation and labeling (Chapter 5); template matching (Chapter 5); laser-vision registration, based on Zhang and Pless' method [Zhang 2004]; semantic and contextual interpretation (Chapters 5 and 6); and the semantic fusion based on MLN (Chapter 6). 21

3.2 Top figure: The perception system ranges from 2 up to 20m in both sensor spaces. Bottom pictures (from left to right): electric vehicle, dual-core computer used to process sensor data in separated threads, a SICK 2D laserscanner (100^o degrees of aperture angle) and a PointGrey camera set to a resolution of 1024 × 768 pixels (45^o degrees of field of view). 22

4.1 Haar-like features prototypes (edge orientations): (a), (b), (c), and (d) are line features; (e) and (f) are edge features, and (g) is the center-surrounded feature. 31

4.2 The sum of the pixels in rectangle D can be calculated with four references: 4 + 1 - (2 + 3). 31

4.3 HOGs are computed densely within the detection windows, in the same way that the Haar-like features. The values of the gradients are accumulated into bins to form the histogram over the cells (right-most image). 33

4.4 CNN architecture: The first 5 neural network layers correspond to the input gray-level image (input layer), C_1, S_1, C_2 and S_2, working as a trainable feature extractor. C layers convolve input image and the previous S layers, which, in turn, subsample C layers, reducing the dimensionality of the feature vector, obtaining only features which are hopefully important to the recognition process. In the last 2 layers, an MLP neural network is used to classify S_2 features. During the training, the error signal is backpropagated up to C_1 layer. 36

4.5 HFI architecture. Three variables are considered in the first layer of the fusion system: rate between the perimeters of the detected windows, Euclidean distance between the two centers of the detection windows divided by the highest detection window width, and the scaled scores of two classifiers; in the second layer, the output for each pair of fuzzy variable feeds another fuzzy system in order to give the final confidence score. Although it is possible to include more classifiers, it would be difficult to manage the exponential growing of the cascade fuzzy rules. 38

4.6 Fuzzy variables and knowledge surfaces: (a), (b) and (d) represent the four fuzzy input variables of the first stage; (c) and (e) are fuzzy output variables of the first stage, which feed the fuzzy system of the second stage; (f) is the fuzzy output variable of the second stage, giving the final fusion confidence. (g), (h) and (i) are the resulting knowledge surfaces of the application of the fuzzy rules over the fuzzy variables. 39

4.7 Initial ensemble. Composed of two feature extractors, HOG and LRF (provided by the CNN), and four classifiers, two MLPs and two SVMs. Abbreviations: HS = HOG/SVM, HM = HOG/MLP, LS = LRF/SVM, LM = LRF/MLP 40

4.8 LRF parameters: Parameters were found by varying the number of C and S feature maps, and the number of hidden neurons. The parameter combination with the best performance was $F_{C_1} = 4$, $F_{S_1} = 4$, $F_{C_2} = 14$, $F_{S_2} = 14$ and $HN = 25$, with a hit rate equal to 94%. 43

4.9 ROCs of the component classifiers: Curves show performance of the individual classifiers over DaimlerChrysler data sets. In (a), performance of MLP, linear SVM, RBF SVM and poly3 SVM over LRF features are depicted; in (b), SVM with linear, RBF and poly3 kernels are plotted over HOG features. 44

4.10 Some samples of DaimlerChrysler data sets (a) and (b), and lighting transformed version (c): From left to right, original image, shadowy and sunny effects. In (d), edge information after the lighting transformation over (c). 45

4.11 Evaluation of lighting transformation applied to DaimlerChrysler validation data sets: (a) and (b) illustrate the performance of the best individual classifiers. 46

4.12 Final ensemble architecture: The component classifiers provide scores with values $[0, +\infty]$, which are scaled to $[0, 1]$ by a logistic link function (LLF). Finally, the Sugeno FI combines the classifier results, providing another confidence score $[0, 1]$, which is thresholded by the minimum value of the combination of two classifiers. 48

4.13 ROCs of the video sequence evaluation: (a), (b) and (c) show the evaluation of different sliding window sizes, considering also strides of 1/8 for the 64x128 pixel window and 1/9 for the others. In (d), ROCs of the evaluation among three types of ensemble: proposed ensemble using Sugeno FI as a fusion method, HFI over Haar wavelet/adaboost + HS, and HFI over LS + HS. 51

4.14 Examples of frame sequences: Annotations (ground truth) are the dark bounding boxes, while the light bounding boxes are the objects detected by the ensemble classifier. In frame 189, there is an example of a false alarm. 52

5.1 Outline of laser-based detection system. Each one of the process frames is detailed in the next sections: **coarse segmentation** is given in Section 5.2.1.1; the β-skeleton procedure in charge of performing the **fine segmentation** is addressed in Section 5.2.1.2; the labeling procedure based on **Procrustes analysis** is described in Section 5.2.2; after finding the labels and respective confidences, a **context analysis** is accomplished according to the procedure given in Section 5.3. 57

5.2 Coarse-to-fine segmentation. In (a), a coarse segment corresponding to a human torso and hand. In (d), there are 2 fine segments, while, in (f), there are 23 fine segments. Note that, from (b) to (f), as β grows, the sparseness of the random graphs increases. 60

5.3 Cluster tendency: For each set of training images, β-skeleton graphs were computed with β varying in the interval [1,3], and stride equal to 0.1. The lower the value of T, the better. As a result, $\beta = 2.1$ was chosen. 61

5.4 Tree of shapes: Pre-shape silhouettes of 10 different persons were averaged to be used as shape references. Back and front models were used with hands close and far from the side of the body, while side models were considered in left and right positions. 63

5.5 Results of laser segmenting and labeling over a testing data set. On the right top and bottom of each image, information about the vehicle speed and frame number, respectively. In (a), two potential occlusions which will be semantically fused later, according to spatial relationships between objects. In (b), a very low confidence in hand labeling will take it to be discarded. In (c), an example of noise segment which is always discarded. In (d), an example of a side torso. 64

5.6 The "nearness" between two segments is defined as the angle (proportional to the laser angle resolution) given by the end and start points of the two segments, relative to the laser sensor. The "nearness" is only aplicable to partial segments in reference to adjacent full segments. 66

5.7 Some examples of occlusions. In (a) and (b), it is shown examples of potential occlusion and arm, "near" a full segment. (c) shows two potential occlusions discarded because they do not satisfy the concept of "nearness". In (d), an example of a wrongly discarded "potential" pedestrian (second last pedestrian to the right); in this last situation, vision may help the detection, since our fusion system is not ROI-based. 67

6.1 Two scenarios where traditional laser and vision fusion systems would fail. The yellow bounding boxes are the result of an image classifier (laser independent), while the blue ones are the result of laser segmentation. In (a), an example of partial segmentation, which is not usually considered by a traditional laser segmentation system, is illustrated; hence, only the pedestrian closer to the camera could be detected by ROI-based approaches. In (b), pedestrians might be considered by the existing fusion methods, only where image classification and laser segmentation match. 70

6.2 A window sized 1.0m × 1.8m is shifted onto horizontal and vertical directions in laser sensing space, with a stride of 0.20m, ranging over 2m up to 20m in depth. The searching area is constrained by the viewport formed by the planes π_1 and π_2 (parallelepiped). 73

6.3 A window is slid in laser sensing space onto horizontal and vertical directions. Only those projected planes lying on the image are used. Depth information of each 3D window is kept for posterior reasoning. 74

6.4 Pedestrian window (54x108 pixels) broken into two main parts: upper body (34 pixels) and lower body (24 pixels), covering shoulder and head, and the waist zone, respectively. 74

6.5 Samples classified by the parts-based HLSM-FINT in the 3D sliding window framework. In (a), some difficult objects are successfully detected by our parts-based ensemble detector, while in (b), it is illustrated some detection failures. 75

6.6 MLN framework. Either in the training or in the inference stages, a set of grounded ML formulas must be generated according to the FOL formulas in Table 6.1. After training, weights are given to each one of the FOL formulas in order to perform inference over the grounded formulas generated by the rules of laser segmentation and image classification. At the end, a probabilistic output is provided for each object found in the scene. 79

6.7 ROC curves of pedestrian detection regarding each sensor space and semantic fusion. The operating points in the curves were chosen to lie on 0.5 FAPF for all curves. (a) Results on laser raw segmentation and using first-order formulas 7 and 8. (b) Image classified with monolithic and parts-based versions of HLSM-FINT. In (c), results of the semantic fusion composed by our parts-based classifiers and all first-order formulas in Table 6.1. 82

6.8 Samples of resulting semantic fusion. In (a) and (b), first rows are the result of the individual classification in each sensor space (the white bounding box is given by an image detector, while the blue one is given by a laser segmentation), and the second rows are the fusion result. In (a), a set of results where the laser fails and the image detector complements the recognition process. Note that the depth information when laser fails is estimated from the 3D searching windows. In (b), a highly dense occlusion scene. 83

6.9 The results of our proposed fusion method applied to those scenes of Figs. 6.1. 85

C.1 A normalized image window runs through all scales and positions over the input frame. The smallest is the scale of the input frame, the closest is the object. 106

C.2 SIMD processing: Adding two vectors VA and VB, sending the result to a third vector VC [Buttari 2007]. 107

C.3 Cell processor design [Buttari 2007]: The architecture is composed by a dual-threaded processor called PPE and 8 vector processors, called SPE. Only 6 SPEs are programable available under Linux. 108

C.4 Fragmentation approach: A single image is fragmented into same sized pieces. Each piece is then sent to one different SPE. 109

C.5 Multi-resolution approach: Each level of the pyramid is sent to an SPE. This approach is feasible as long as each scaled image can fit into the LS of each SPE. Otherwise, a multiple step technique should be implemented to process each image, considering multiple references to the LS. 110

C.6 Framework of the parallel HOG. This is comprised of 3 modules: In module (1), the input image is broken into n octaves (scaled input image); for each octave, the image is broken into smaller pieces to be sent to the SPUs. In module (2), the smaller pieces of an image come in parallel, while in each SPU a piece of code is responsible for extracting HOG features, posteriorly classifying them with a linear SVM. Finally, in module (3), the results are shown. Modules (1) and (3) run in PPU, while module (2) runs in SPUs. . 111

List of Abbreviations

ANN	Artificial Neural Network
CBE	Cell Broadband Engine
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
DET	Detection Error Tradeoff
DMA	Direct Memory Address
EEC	Exponential Error Count
EIB	Element Interconnection Bus
FAR	False Alarm Rate
FI	Fuzzy Integral
FOL	First-order Logic
FOV	Field of View
FPS	Frames per Second
GLB	Gabor Local Binary
GMM	Gaussian Mixture Model
HFI	Hierarchical Fuzzy Integration
HN	Hidden Neurons
HOG	Histogram of Oriented Gradients
HR	Hit Rate
IAV	Inscribed Angle Variance
IID	Independent and Identically Distributed
ITS	Intelligent Transportation Systems

KB	Knowledge Base
LEM	Laplacian Eigen Map
LIDAR	Light Detection and Ranging
LRF	Local Receptive Fields
LS	Local Store
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MFC	Memory Flow Controller
MLN	Markov Logic Network
MLP	Multi-layer Perceptron
MRF	Markov Random Fields
MSR	Multi-scale Retinex
MV	Majority Vote
PCA	Principal Component Analysis
PPE	Power Processing Element
PPU	Power Processing Unit
RBF	Radial Basis Function
RF	Receptive Field
ROC	Receiver Operating Characteristics
ROI	Regions of Interest
SIFT	Scale-invariant Feature Transform
SIMD	Single Instruction Multiple Data
SLT	Statistical Learning Theory

SMO	Sequential Minimal Optimization
SPE	Synergistic Processing Unit
SPU	Synergistic Processing Unit
SRM	Structural Risk Minimization
SVM	Support Vector Machines
TDNN	Time Delay Neural Network
UTA	Urban Traffic Assistant
VC	Vapnik-Chervonenski
VMX	Vector Multimedia Extension

Part I

Presentation

Chapter 1

Introduction

Contents

1.1 Motivation 4

1.2 Goals 5

1.3 Key contributions 6

1.4 Chapter map 7

> *Those who work in the field of artificial intelligence cannot design a machine that begins to rival the brain at carrying out such special tasks as processing the written word, driving a car along a road, or distinguishing faces. They have, however, shown that the theoretical difficulties in accomplishing any of these tasks are formidable.*
> David Hubel, in “Eyes, Brain and Vision”.

The use of perception machines has been growing everyday, leading us to believe they will be mandatory in the transportation systems in the near future. Actually, in the last decades, several researchers have been developing complete perception architectures for intelligent transportation systems (ITS) [Reisman 2004], [Leibe 2005], [Leibe 2007].

Amongst the various tasks that an intelligent perception system is expected to perform, object detection could be considered a primary goal to provide safe and reliable vehicle autonomy or assistance (refer to [Gandhi 2007], for a recent survey in collision detection applications). Over the plethora of world objects which a human being learns how to avoid in driving situations, the human being himself is one of the most difficult objects for an intelligent machine to detect. This is so because a person can appear in several poses, positions, forms, sizes, and colors.

Let us consider a vision-based object detection system and the Fig. 1.1. Probably, most of us would have no difficulty in recognizing the two persons under the sunset.

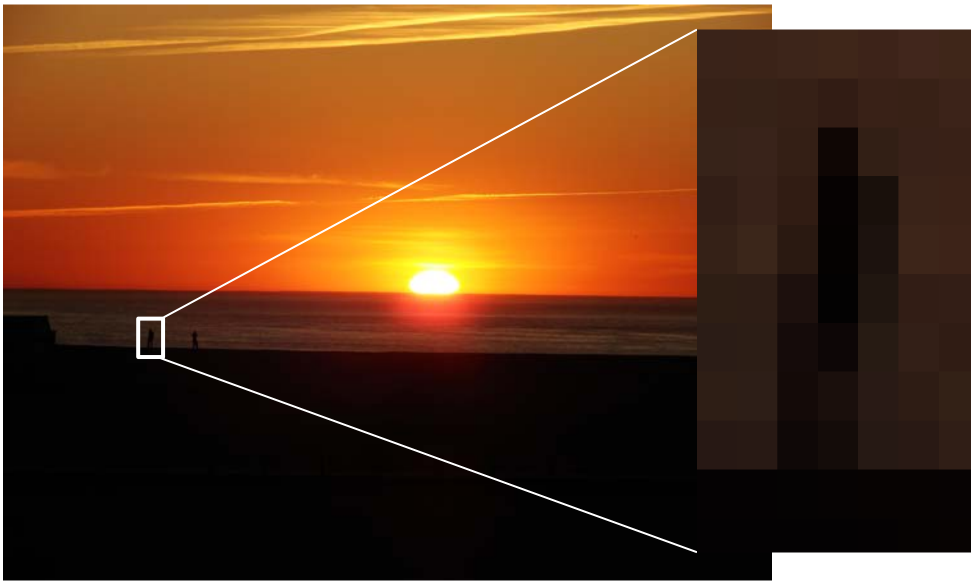


Figure 1.1: Most of us could identify the two persons under the sunset. But for an image detector, it could be cumbersome to do it, as the persons present either very low resolution in the original image, or they loose their inherent aspect after resized (sided image).

However, it could be cumbersome for an image detector, for example, to identify the persons in their small current sizes, or even after amplifying them, because they loose their inherent aspect.

This example in Fig. 1.1 is just one among various situations which a perception system must deal with, to be consistently employed in dynamic outdoor scenarios. As the problem is not easy, a synergism of methods, sensors and architectures is essential to get as much information as possible in order to make, at least, a coherent decision.

After this short and introductory presentation of the topic, motivation, goals, key contributions, and chapter map of this thesis will be presented in the next sections.

1.1 Motivation

In a complete object detection framework, there are multiple problems to be addressed, such as:

- **Object segmentation or searching.** The goal is to segment or, ultimately, to search for regions of interest (ROI), in order to find an object within a cloud of raw points. If an image sensor is used, for example, segmentation can be done at the pixel-level, or by shifting a standardized window over many image positions and scales (some other alternative methods, which avoid a greedy search, can be found in [An 2009]). In planar range finders, on the other hand, gradient methods are one alternative among many techniques [Borges 2000].

- **Object detection.** It aims at the recognition of an object of interest given an input of raw data. It begins with object feature extraction, while the complete goal is achieved by a classifier in charge of naming the input feature vector as object or non-object.

- **Object tracking.** Tracking an object through its possible trajectory is important for two reasons: To eliminate possible inconsistencies in object recognition level, and to analyze object behavior, providing decisions for vehicle path planning and navigation, for example. Several problems have to be addressed in a tracking system, such as: initialization, registering of objects among N consecutive sensor readings, crossing objects, and so forth. A comprehensive survey in visual tracking can be found in [Yilmaz 2006].

Each of those items could certainly be a research topic by itself. Therefore, the scope of this thesis resides essentially in object recognition. We are particularly focused on the problem of multi-sensor object detection, motivated to answer the following question: *Where is a pedestrian in the scene?* Whereas we are able to change the object of the question for any other one (e.g., vehicles, poles, trees, animals, balls) [Oliveira 2008b], pedestrians were chosen as the object of interest because they represent a challenging task, giving rise to developing a life saving system. Despite that question has been in the center of many scientific proposed methods, a definite answer seems far from being found. Our challenge is thus to conceive a detector with a state-of-the-art performance by building it on a synergistic framework. To carry it out, some researchers [Torralba 2003], [Hoiem 2006] have demonstrated that extracting contextual information is of fundamental importance to reach the ultimate goal – having object detection systems in daily use.

1.2 Goals

One of the first object detection systems using vision was proposed by Papageorgiou and Poggio [Papageorgiou 2000]. The goal of this single detector was to be incorporated in the DaimlerChrysler Urban Traffic Assistant (UTA) to recognize pedestrians in urban scenarios. Since then, many other methods have been conceived to tackle that problem. As a consequence, ways to assess the performance of the proposed methods have become of great importance. For example, Papageorgiou and Poggio assessed the performance of their system using receiver operating characteristics (ROC) curves, but nothing is said about the number of images used to analyze the detector performance. More recently, Dollar et al [Dollar 2009], by using a data set with approximately 250 000 frames, demonstrated that an extensive evaluation of a detection system is essential if our goal is a real-life application. This way, our main goal is then not only to conceive a pedestrian detection system, but also to evaluate it thoroughly, demonstrating the performance of our proposed method in real scenarios. To this end, we started evaluating our local detectors by using standard data sets for comparison reasons. Finally, we assessed the overall performance of the whole system over gathered data sets, having characteristics of object occlusion, shadow covered areas, and lighting change – similar to those presented in real-life scenarios.

Other specific goals are:

- **To increase the detection rate of existing local detectors by ensemble of classifiers.** As we mentioned before, due to the complexity of the problem, by using only one pair of feature extraction-classifier, it is difficult to deal with all dynamic situations. By designing ensemble of classifiers, we are able to overcome many problems encountered by single detectors.

- **To get object localization.** There are many ways to recover object localization by using multi-sensor fusion. Although stereo vision systems are the main alternatives, the use of 2D light detection and ranging (LIDAR)[1] along with vision is not yet fully explored. This way, one of the main goals of this work is to integrate these two sensors more robustly.

1.3 Key contributions

Our key contributions are listed bellow.

- **Ensembles of classifiers.** Our first aim was to develop a single local detector. While going deeper into this study, the limitations of this type of detector were becoming apparent [Araujo 2008]. This way, we decided to investigate new paradigms to integrate ensemble of classifiers, initially proposing a method coined as hierarchical fuzzy integration (HFI) [Oliveira 2007a],[Oliveira 2007c]. Posteriorly, we

[1] Sometimes we refer to this type of sensor as just laser scanner, or simply laser, used interchangeably with the term LIDAR along this document.

realized that this system was limited due to twofold drawbacks: The use of few classifiers, because of the exponentially growth of fuzzy rules, and the performance of the overall system did not lead to state-of-the-art classification rate, even with high performance single classifiers as its components. As a result, we proposed a second approach [Oliveira 2007b], [Oliveira 2008b], [Oliveira 2010b], HLSM-FINT (stands for HOG, LRF, SVM, MLP, Fuzzy Integral) based on synergistic elements in its architecture, extensively studied, analyzed (also with respect to sensitivity to lighting conditions), and compared to other methods.

- **A featureless approach for laser segmentation and labeling.** Rather than following the previous approaches for object detection using LIDAR based on geometric feature extraction and classification, we proposed a clustering-based method using β-skeleton random graphs and a template matching to score the labeled segments, using Procrustes analysis. This method deals also with occlusion by partial segmentation within a context-aware framework [Oliveira 2010a].

- **A semantic fusion of laser and vision.** Existing architectures of laser and vision rely on ROI extraction over laser points, following by an object image classification, or, on the other hand, on extraction of features in both sensor spaces. Both methods are defined on independent and identically distributed (IID) classifiers (or fusion systems). All methods are ROI-based and fail whenever laser does. We propose a different approach, based on semantic integration with contextual information and no need of IID assumptions, which can be endowed with the vision system even if the laser fails [Oliveira 2010c].

1.4 Chapter map

The remainder of this thesis is organized as follows.

- **Chapter 2**. Previous work is presented and discussed. The goal of this chapter is to present the background of our work with respect to shared, extended or novel ideas from existing methods.

- **Chapter 3**. Along with Chapters 1 and 2, this one represents the essential background to comprehend our proposed work. The outline of our multi-sensor detection system is given, and the characteristics of our laser and vision (gathered) data sets are summarized.

- **Chapter 4**. The single classifiers used in our proposed ensemble detectors are presented according to our implementation. Also, this chapter describes our two proposed ensembles of local detectors, HFI and HLSM-FINT, assessing their performances on a thorough analysis over several data sets.

- **Chapter 5**. The constituent parts of our laser detection system are described: coarse segmentation, fine segmentation, labeling, and contextual reasoning. These components were built on the idea of dealing with occlusion in urban scenarios.

- **Chapter 6**. This chapter ends the description of the proposed pedestrian detection briefly addressed in Chapter 3. We present a parts-based version of our HLSM-FINT detector. This is applied to constrained sliding windows originally shifted in the laser sensing space, and posteriorly projected in the image space after sensor registration. The semantic multi-sensor fusion is introduced. Finally, the performance of the sensor-driven detectors and the proposed fusion method is assessed over data sets gathered in challenging urban scenarios with frequent object occlusion.

- **Chapter 7**. A discussion about the proposed framework, its limitations, and aspects of possible on-the-fly implementation are discussed. Some future directions are suggested.

- **Chapter 8**. Concludes the thesis.

Chapter 2

Background

Contents

2.1 Laser-based object detection 9

2.2 Object detection in images 10

2.3 Ensemble of classifiers 12

2.4 Fusion of laser and vision 13

2.5 Data sets 14

2.6 Relation to our work 15

For since God has given to each of us some light with which to distinguish truth from error, I could not believe that I ought for a single moment to content myself with accepting the opinions held by others unless I had in view the employment of my own judgment in examining them at the proper time.
René Descartes, in "Discourse on Method"

This chapter introduces some background in the field of object detection. Each section bellow addresses a particular topic inside our object detection framework. In the last section, we describe the relation of each previous work to our proposed method.

2.1 Laser-based object detection

Object detection using laser usually relies on geometrical feature extraction over laser points. Features are then classified with a generative or discriminative method, or they are simply analyzed to match some criteria.

Song et al. [Song 2002] propose a generic detection system using a laser. They use a line-based segmentation along with a set of geometrical primitives (lines, circles and ellipses), which are fitted into the segments to name it as object or non-object. Premebida and Nunes [Premebida 2005] review similar methods for feature extraction and segmentation. They also propose a Gaussian mixture model (GMM) based classification system to detect pedestrian in outdoor environments [Premebida 2006]. Xavier et al [Xavier 2005] propose a new type of feature called inscribed angle variance (IAV), which performs a leg-segment feature correspondence in indoor environments. Zhang et al. [Zhang 2003] introduce two new algorithms to extract geometrical features based on the Gaussian-newton algorithm and an online multiple model filtering for indoor classification. Arras et al. [Arras 2007] use a set of statistical primitives such as number of points, standard deviation, mean average deviation from median, and eleven more, proving them as an input to an adaboost classifier. The goal is to classify leg segments as person or non-person, in indoor environments.

2.2 Object detection in images

To accomplish the task of detecting objects in images, it is necessary to find a way to represent this object beyond the raw pixels, since they are not representative enough to be used directly. A deeply review of the literature in this field suggests that many feature representation methods have been introduced in the last decade, while classification systems continue relying on standard methods of machine learning, such as: multi-layer perceptron (MLP), support vector machine (SVM), and boosting. While the first two methods are based on the same foundations (differing from each other in the way of training), the boosting methods are usually employed when it is necessary to have a faster computational approach. Another aspect concerning boosting-based classifiers is that they are very useful when the input features do not lie on a vector space [Tuzel 2008], where standard MLPs[1] or SVMs are not adequate to be applied. Finally, it is worth noting that SVM-based classifiers usually present a better performance [Kecman 2001], among those classifiers. This is so because in its training procedure, a convex and quadratic optimization problem is cast, which leads the solution to be global. SVM is a frequent choice in object detection field [Papageorgiou 2000], [Dalal 2005], [Munder 2006], [Llorca 2006].

One of the first detection systems, based on Haar-like features, was proposed by Papageorgiou and Poggio [Papageorgiou 2000]. That type of feature was used as an input into a SVM framework. Another early detection system was proposed by Wohler et al. [Wohler 1998], and was based on a time delay neural network (TDNN). Instead of inputing single feature vectors, TDNN is fed by a sequence of raw frames, posteriorly

[1] In [Porikli 2006], a special MLP is proposed to deal with covariance features.

extracting shapes in a trainable convolutional layer in the network. Viola and Jones [Viola 2001], motivated by Papageorgiou and Poggio's feature extractor, built a new classification system. Speeding up not only the way of extracting the Haar-like features, by means of an integral image technique, but also the classification of those features by using an adaboost classifier, Viola and Jones were able to achieve a detector 15 times faster than any other previous method in face recognition, at that time. Although, Haar-like features with adaboost classifier have become the very choice for standard vision systems in ITS, those features present quite a poor performance for pedestrian detection in cluttered scenarios. The reason for that is that Haar-like features often fail to capture object representation when the contrast between object and background presents small variations. To solve this limitation on discriminating small object-background contrast, Dalal and Triggs [Dalal 2005] propose to use a histogram of oriented gradient (HOG) feature extractor based on the scale-invariant feature transform (SIFT) descriptor proposed by Lowe [Lowe 1999]. Dalal and Triggs' HOG descriptor is applied densely, in contrast to the SIFT approach, presenting superior performance in comparison to Haar-like features, principal component analysis (PCA) applied to SIFTs and shape contexts, over challenging data sets.

As many detectors have been designed, the need of standard benchmark procedures has been mandatory to assess classification performance. This way, Munder and Gavrila [Munder 2006] propose an extensive study on DaimlerChrysler data sets gathered in urban scenarios. The analysis of that authors included a set of feature extractors such as PCA, Haar wavelet (also called Haar-like features), and a type of LRF obtained by the weights of the MLP's hidden layer over PCA and Haar wavelet features. They conclude in their work that the latter approach, classified by SVM or Adaboost, provides the highest performance (with the Adaboost presenting lower computational cost). Further, Tuzel et al. [Tuzel 2008] also used DaimlerChrysler data sets to compare their proposed detector based on covariance matrices. Covariance matrices do not lie on vector spaces, as it is not convenient to classify them using MLPs or SVMs, as shown in [Forstner 1999]. As a result, Tuzel et al. [Tuzel 2008] used a logiboost method. Recently contributing to benchmark the performance of pedestrian detectors, Dollar et al. [Dollar 2009] present a deep study with HOG/SVM classifier over a data set of approximately 250 000 frames. They showed, for example, that the performance of classifiers computed over ROC and based on false alarm per window is flawed, and can fail to predict per-image performance. They also discuss other biased situations in assessing detection performance.

2.3 Ensemble of classifiers

Designing a single feature extractor-classifier, able to cope with a large image variability, is still an open problem. This is so because it is particularly difficult to build a feature extractor that represents uniquely the object of interest. Therefore, the fusion of classifiers has been studied in the last few years with the goal of overcoming certain inabilities of single feature extractor-classifier. One of the main goals of ensembles of classifiers is to explore the diversity of the component classifiers in order to enhance overall classification performance. In other words, since there is no perfect individual classifier yet, by combining them, one can complement the other. If there are errors in the ensemble components, it is expected that they occur in different image objects in order to give rise to fusion methods that improve the performance of the whole system. Furthermore, the rationale of building ensembles is also that most individual classifiers agree in a certain way such that the whole system can be more successful than its component parts. Figure 2.1 illustrates these aspects for a simple ensemble composed of two classifiers.

There are many ways to integrate features and classifiers. Kuncheva [Kuncheva 2004] presents some methods to measure diversity of classifier ensembles. Aksela and Laaksonen [Aksela 2006] describe an in-depth study about correlation between ensemble accuracy and diversity. Several types of component selection criteria for ensemble classifiers are examined in that study. Although there is no unquestionable diversity measure for choosing the best ensemble from an initial set of classifiers, by exploiting the diversity of errors across the component classifiers, it is possible to choose a set of classifiers whose decisions are better integrated by the fusion method. Actually, in [Aksela 2006], it is suggested that the choice of a diversity measure, to prune the classifiers in one ensemble, should rely on the goal of the fusion method (e.g., if the fusion method works on weak or strong classifiers).

Regarding fusion methods for object detection in ITS, Llorca et al [Llorca 2006] addressed a combination of classifiers based on a parts-based ensemble method, by evaluating which classifier performs the best for each part of the human body. Nanni and Lumini [Nanni 2008] conceived a novel ensemble of classifiers based on Laplacian eigen map (LEM) and Gabor local binary (GLB) features, with each feature vector being classified by an SVM. The outputs of each classifier are integrated into a majority voting method using a sum rule. They followed a parts-based approach, where each input image is divided into half parts (top and bottom), each one being represented by LEM or GLB feature vectors.

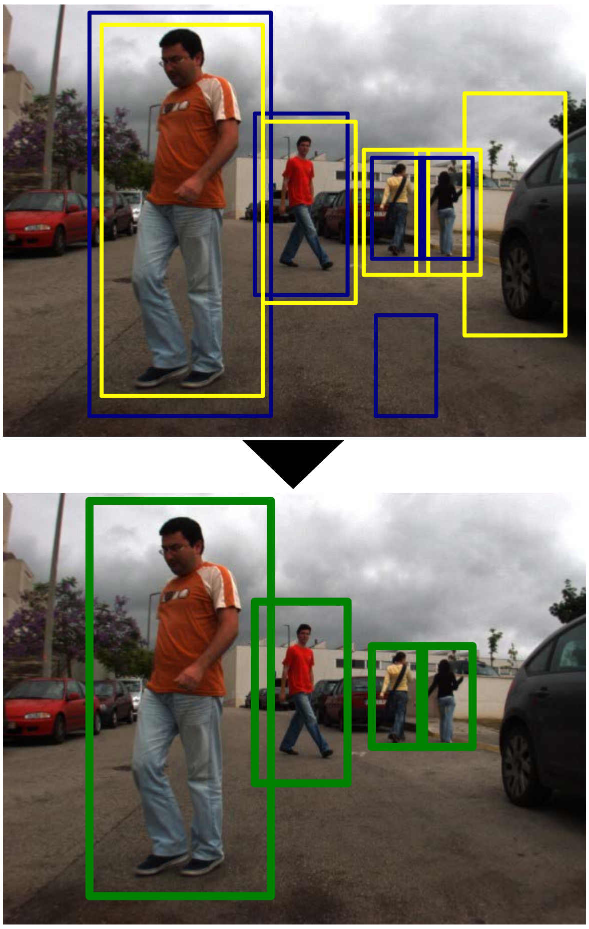


Figure 2.1: Diversity in ensemble of two classifiers: Image on the top shows the individual classification by two classifiers, while image on the bottom is the result of a classification fusion. It is expected that a fusion method takes the complementarity of the component classifiers into account in a way that the performance of the ensemble is superior to the performance of the individual classifiers.

2.4 Fusion of laser and vision

So far, fusion of laser scanner and vision sensors has been performed by assuming that the probability to find an object is identical and usually independent, in both sensor spaces. There are two main approaches for this type of fusion:

- A laserscanner segmentation method is used to find most likely regions of interest (ROIs), where an image classification system is applied.

- IID integration of sensor-driven classifiers (posterior probabilities), or sensor-driven features.

Some of the works based on these methods are presented as follows. Szarvas et al. [Szarvas 2006] rely entirely on laser ROIs in order to find probable areas where a pedestrian might be. Each image projected ROI is classified by a convolutional neural network (CNN) [Lecun 1998]. When the laser fails, no pedestrians are detected. Broggi et al. [Broggi 2008] propose an on-spot pedestrian classification system, which first uses a laserscanner to determine areas between vehicles, and then a Haar-like feature/adaboost classification system to name the projection of these areas in the image. Mahlisch et al. [Mahlisch 2006] propose a spatio-temporal alignment to integrat the laserscanner and the monocular camera. For that purpose, features in both sensor spaces are extracted, feeding a Bayesian classifier to detect cars. Douillard et al. [Douillard 2007] propose an approach similar to the previous one, but rather than a Bayesian rule they use a conditional random field (CRF). Additionally, they are able not only to classify laser and image features, but also to find a temporal relationship between features of sequential sensor readings. Premebida et al. [Premebida 2007] propose to classify features in laser space with a Gaussian mixture model (GMM) while Haar-like features are extracted over the image objects and classified by an adaboost. The confidence scores of the classifiers feed a Bayesian rule to make the final decision. The goal is to classify vehicles and pedestrians.

2.5 Data sets

As mentioned before, it is necessary to standardize benchmarking of classification systems. ROC, F1 score, recall-precision and detection error tradeoff (DET) curves became the standard to assess the performance of a classifier. To evaluate a single or a set of classification methods, common data sets need to have a unique and known source of inputs. Although there is no standard method that guarantees if a given data set is easy or difficult, by making them available online for academic proposes, the scientific community can guarantee some standardization in benchmarking procedures.

Amongst a wealth of data sets for pedestrian recognition on images, some can be considered as difficult or extensive, such as: INRIA [Dalal 2005] and DaimlerChrysler [Munder 2006] data sets. The former data set are composed of images fromGRAZ[2], personal collection and google images. The latter data set were collected from a driving urban car. The characteristics of these data sets are depicted in Table 2.1, while in Fig. 2.2 some examples illustrate each data set. For laser scanner, to the best of our knowledge, there are no reliable data sets available online.

[2] http://www.emt.tugraz.at/∼pinz/data/GRAZ_01

Table 2.1: Some available pedestrian data sets for benchmarking

Reference	**Training**	**Validation**	**Characteristics**
DaimlerChrysler [Munder 2006]	3 data sets of images, each one composed of 4800 pedestrians and 5000 non-pedestrians	2 data sets of images: each one composed of 4800 pedestrians and 5000 non-pedestrians	Composed of 800 pedestrians in 6 poses. 18×36 pixel cropped
INRIA [Dalal 2005]	2478 pedestrians and 12180 non-pedestrians	Frames containing 1805 pedestrians	Images from: GRAZ personal collection, and google images.

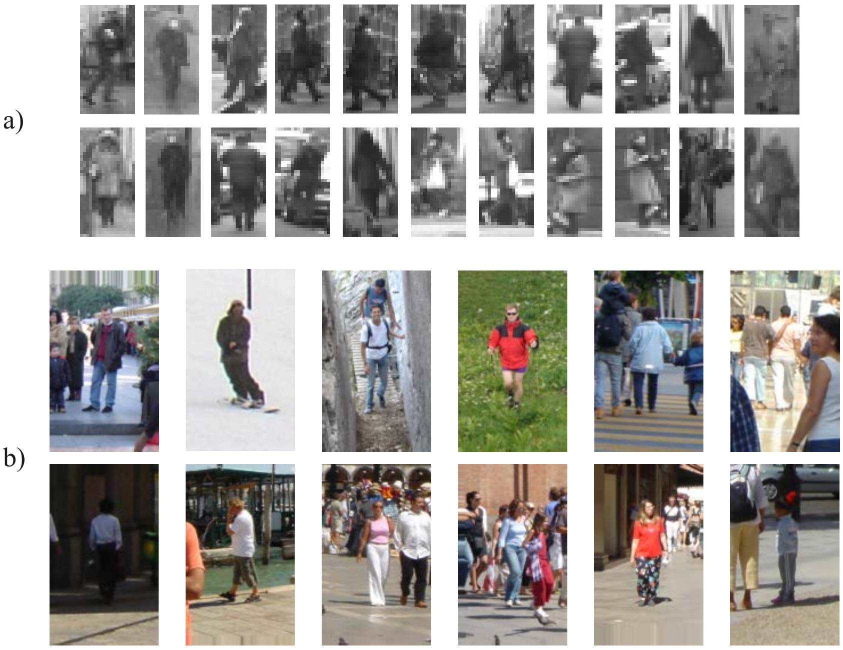


Figure 2.2: Some samples of (a) DaimlerChrysler and (b) INRIA data sets.

2.6 Relation to our work

All of these previous ideas motivated our research to a certain level. For object detection using laser, all cited works deal with geometric feature extraction, classifying the features by a generative (e.g., GMM [Premebida 2005]) or discriminative (e.g., adaboost [Arras 2007]) classifier, or simply taking decisions according to some structure matching [Zhang 2003]. All of them rely on leg-based classification. Instead, we proposed a clustering-based segmentation with subsequent template matching procedure. Laser is mounted in the waist level. A coarse-to-fine segmentation method builds a parts-based segmentation from raw points to semantic parts. A Procrustes analysis is performed on each segmented part to give a level of confidence to match our tree of templates. This approach was particular useful to deal with partial segmentation, providing a spatial relationship of one segment with respect to others in the scene. This method allowed us to treat occlusion robustly.

Regarding object detection in images, we borrowed many ideas from single detectors, always considering that they are limited to certain aspects of feature representation. In this regard, we built our proposed ensembles of classifiers on powerful components, demonstrating that we can achieve a synergism among them in the face of adverse circumstances, like lighting change. Therefore, we use two existing feature extractors, HOG [Dalal 2005], and one based on a local receptive field (LRF). The former was implemented from scratch, providing a flexible way to change its structure, anytime. The latter was used by Wohler et al. [Wohler 1998], in the form of a TDNN; by Munder and Gavrila [Munder 2006], with an MLP applied on Haar-wavelet and PCA feature extractors; and, finally, by Szarvas et al. [Szarvas 2006], in their multi-sensor fusion scheme, following the architecture of a CNN proposed by Lecun et al. [Lecun 1998]. Our LRF approach was based on CNN approach, but instead of a partial connection in the network structure, we used full connection. To the best of our knowledge, Szarvas et al. [Szarvas 2006] were the pioneers of CNN application in pedestrian recognition, however, no analysis about the best structure and parameters was provided. In this thesis, we provide a thorough evaluation with respect to CNN structure and performance over lighting changes for pedestrian detection [Oliveira 2010b]. Using these features, we first propose a classification fusion method based on fuzzy logic, called HFI. A second ensemble architecture proposal is done to overcome some drawbacks of the first method, lately based on fuzzy integral, showing better performance.

The way of ensembling classifiers is crucial to its success. Therefore, we analyzed many metrics presented in [Kuncheva 2004] and [Aksela 2006] in order to choose our ensembles. The latter reference proposes a measurement which best fits our goal of integrating high-performance, single classification systems. This measurement is based on a exponential error count (EEC), counting the importance of the component classifiers not making the same errors too often, with respect to the correct classification.

The evaluation of our final proposed detector was carried out over DaimlerChrysler data set, following the same methodology as in [Munder 2006], additionally considering artificial lighting changes applied to the validation data sets. As a result, we found that our final ensemble detector outperformed the methods analyzed in [Munder 2006] and proposed by [Nanni 2008].

One of the characteristics of DaimlerChrysler data sets is that its pedestrians are always up-right and never occluded. These data sets have 19 600 images for validation (see Table 2.1). Note that considering an application of a sliding window technique, shifting 2 000 windows per frame in a 1 000 frame-wise data set, means that a classifier will be applied on 2 million cropped images. In this regard, in order to present an in-depth evaluation of our proposed method, we additionally evaluated our ensemble detector over our own data set[3], named “ISR-UC-imgs”. For that, we trained our ensemble detector with INRIA training data set plus a set of extra images collected via a bootstrapping method.

Even though a thorough analysis of our ensemble detector has been carried on our ensemble detector, the “ISR-UC-imgs” data set was still missing frequent occlusion situations. Therefore, we collected a new data set, in a more systematic way to include occlusion and more challenging scenes, using laser and vision. These data sets were called “ISR-UC-imglidar-sync”. Over these data sets, our monolithic ensemble detector achieved poor performance, leading us to apply it in a parts-based framework. This improved its performance, but it still did not reach the expected hit rate. As a result, we decided to integrate our parts-based ensemble detector to a laser, finally proposing our semantic sensor fusion.

Our sensor fusion was conceived to overcome problems with existing laser-vision architectures, since they have been designed by assuming that the probability to find an object is identical and usually independent, in both sensor spaces. These assumptions limit the way that laser and vision are integrated, mainly when the laser fails. This led us to formulate a new framework to incorporate semantic primitives and contextual information. To this end, Markov logic network (MLN) [Richardson 2006] was used as the final fusion system, after processing the object attributes on many levels.

[3] The characteristics of this data set will be presented in the next chapter.

Chapter 3

Multi-sensor object detection in urban scenes

Contents

3.1 Proposed framework outline **19**

3.2 Experimental setup and collected data sets **20**

3.2.1 Gathered data sets 22

3.3 Remarks **23**

The world is but a canvas to the imagination.
Henry David Thoreau

In this chapter, we describe the outline of the proposed framework, presenting the main concepts behind each part. In addition, the experimental setup and the characteristics of the gathered data sets, used to evaluate the multi-sensor detection system, are also provided.

3.1 Proposed framework outline

The proposed method was conceived to be used as the perception system of a low-speed electric vehicle, which will be employed for autonomous people transportation in specific zones like historic city centers. The main goal of this work is thus to provide sensing accuracy rather than real-time processing, which can later be achieved by a special parallel implementation [Oliveira 2008a].

Figure 3.1 depicts the proposed framework, which is comprised of 6 main blocks: The image object detection based on our proposed ensemble detector (HLSM-FINT);

laser segmentation and labeling; template matching; a laser-vision registration procedure, proposed by Zhang and Pless [Zhang 2004]; semantic and contextual interpretation; and semantic sensor fusion based on MLN. The ensemble detector is described, along with our early classifier ensemble (HFI), in Chapter 4 (we describe the parts-based version of HLSM-FINT in Chapter 6); laser segmentation and labeling, and the template matching procedure, based on Procrustes analysis, are detailed in Chapter 5; finally, in Chapter 6, we describe the integration of our parts-based detectors, applied in each sensor space, within our proposed semantic fusion based on Markov Logic Network.

In laser space, we use a featureless approach, based on a clustering method. It starts finding coarse segments, $\{c_n\}$, where $n = 1, ..., N$, posteriorly sub-segmenting each c_n into fine segments $\{f_m\}$, where $m = 1, .., 3$. This latter step is done by a β-skeleton random graph. The laser is mounted at the pedestrian-waist level (see Fig. 3.2), avoiding common problems as those observed in leg-level mounted systems. Therefore, the fine segmentation step expects at most 3 body parts (2 arms and the torso). Each of these parts is then labeled, contextually interpreted, and finally modeled by an MLN.

Rather than applying sliding windows directly to the image, we decided to use the geometry of the laser reference frame in our favor. For that, windows are shifted in 3D space, onto horizontal and vertical directions, and subsequently projected into the image by the registration procedure. For those 3D sliding windows, it is assumed that the laser is always parallel to the ground. In the last stage, all information is processed according to first-order clauses, subsequently structuring a Markov random fields (MRF) with a grounded first-order formula in each node, which is so called a ground MRF. At the end, marginal probabilities are then computed and outputted for each test case.

3.2 Experimental setup and collected data sets

Our sensor apparatus was mounted on an experimental vehicle, depicted in Fig. 3.2, which shows the spatial arrangement of the sensors. A SICK 2D laserscanner was mounted at a distance of 0.9m from the ground, preventing segments with small number of points from being easily discarded at the segmentation step. This problem usually happens in leg-level mounting. A PointGrey camera sensor, set to a resolution of 1024×768 pixels, was mounted on the top of the vehicle's roof.

The sensing system ranges from 2 up to 20 meters, providing enough time to stop the low-speed vehicle whenever a pedestrian is detected in critical areas. The maximum speed of the vehicle is approximately 30km/h. The laser scanner was set with an aperture of

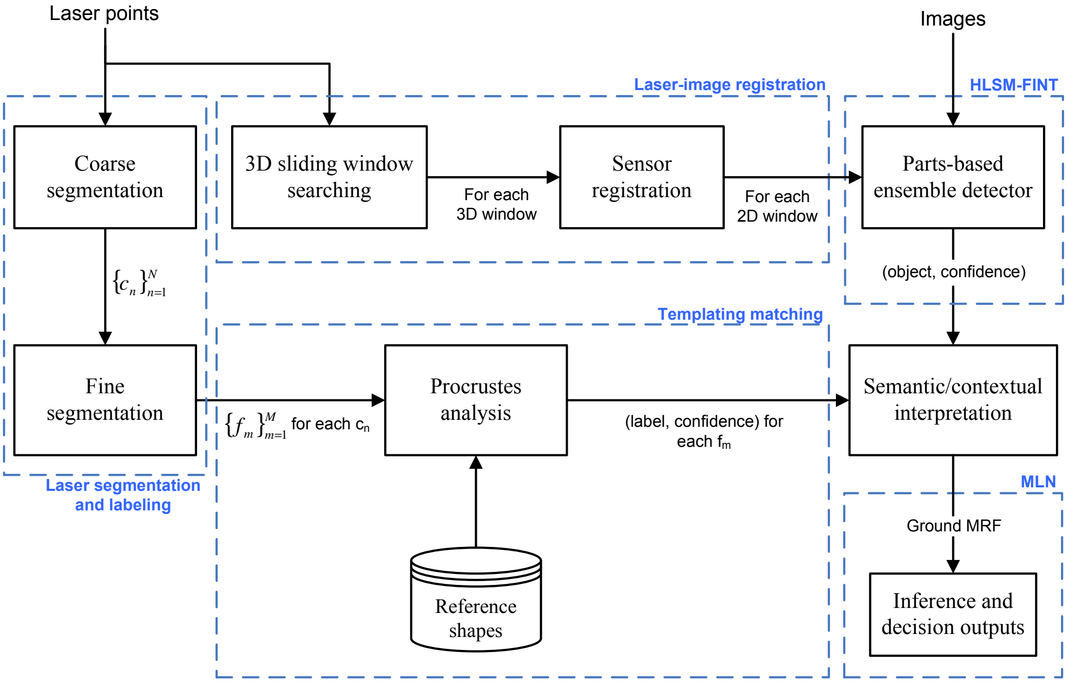


Figure 3.1: The proposed framework is composed of 6 main blocks: The image object detection based on our parts-based ensemble detector HLSM-FINT (Chapters 4 and 6); laser segmentation and labeling (Chapter 5); template matching (Chapter 5); laser-vision registration, based on Zhang and Pless' method [Zhang 2004]; semantic and contextual interpretation (Chapters 5 and 6); and the semantic fusion based on MLN (Chapter 6).

100^o degrees, while the camera forms a field of view (FOV) of 45^o degrees. The resolution between laser beams is 0.25^o degrees. Some pictures of our equipped vehicle and sensors are available in the bottom line of Fig. 3.2.

For each sensor is assigned a thread in a dual-core computing system with CPU affinity (`pthread_setaffinity`) and synchronization barriers (`pthread_barrier`), using POSIX kernel APIs. Sensor data were acquired and saved entirely in the main memory during acquisition time, being saved to the disc only at the end of the sensor acquisition. This procedure was performed to guarantee data synchronization of the two sensors, acquired at a rate of 15 frames per second (FPS), being 15 fps in laser acquisition[1], and 20 fps in camera acquisition[2].

[1] This frame rate was obtained with a special laser driver found in http://sicktoolbox.sourceforge.net/.

[2] Synchronization of data sensors is performed by means of synchronization barriers, which enforces the final rate to be equal to the slower sensor.

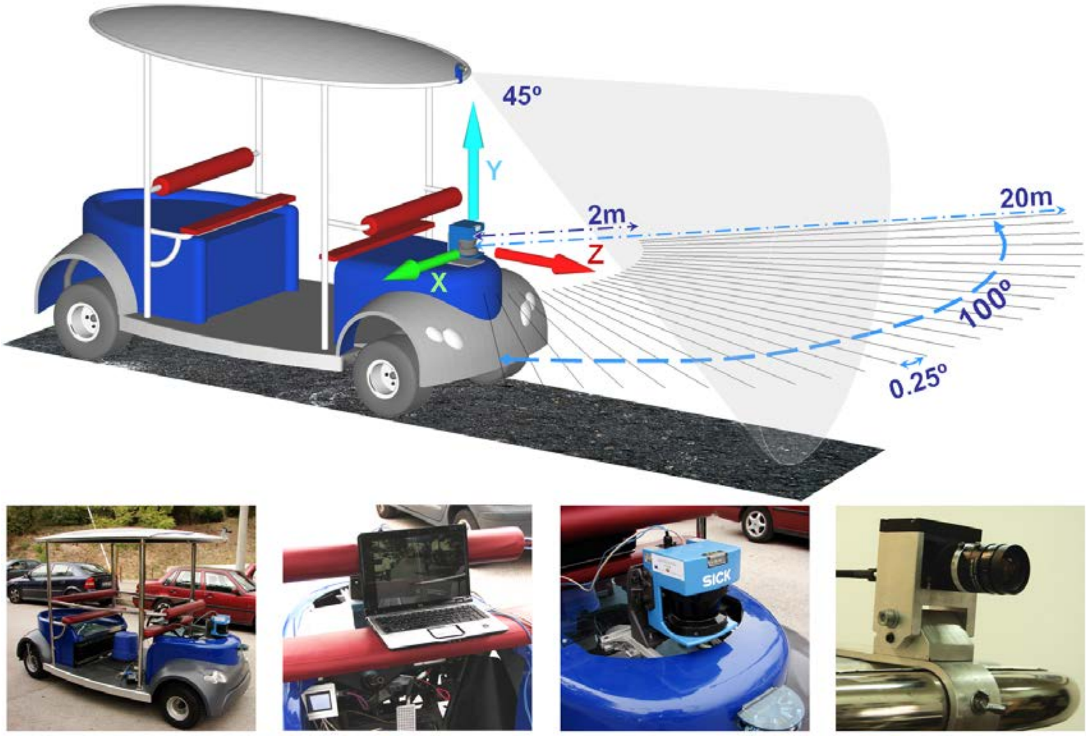


Figure 3.2: Top figure: The perception system ranges from 2 up to 20m in both sensor spaces. Bottom pictures (from left to right): electric vehicle, dual-core computer used to process sensor data in separated threads, a SICK 2D laserscanner (100^o degrees of aperture angle) and a PointGrey camera set to a resolution of 1024 × 768 pixels (45^o degrees of field of view).

Table 3.1: Characteristics of "ISR-UC-imglidar-sync" data sets

Sequence	**# Frames**	**# Annotated objects**	**Dist**
#1	2 672	3 429	160 m
#2	2 157	3 692	160 m

3.2.1 Gathered data sets

The electric vehicle was driven through our campus to gather the data sets. In the traveled trajectory, several pedestrians walked in front of the vehicle in several manners, individually or in groups. The collected data sets are characterized by shadow covered areas, many degrees of object occlusion, illumination changes, many cars in the margins of the road, and few frames without pedestrian presence. Each image was manually annotated to include all objects in the human field of view (hard annotation) only considering objects in the range of 2 up to 20 meters. These data sets were called "ISR-UC-imglidar-sync", and their characteristics are summarized in Table 3.1.

Table 3.2: Characteristics of “ISR-UC-imgs” data sets

Sequence	**# Frames**	**# Annotated objects**
#1	486	187
#2	454	747
#3	364	430

3.3 Remarks

The spatial arrangement of the laserscanner and camera was structured to facilitate not only the contextual priors in the image searching procedure, as described in Chapter 6, but also avoiding laser segmentation problems with leg-level mounting. Actually, in previous setups, we experimented some of these difficulties, namely when using the SICK laser at leg level, and the camera on the top of this sensor. The data sets called “ISR-UC-imgs” were acquired with that early setup, initially intended to be image and laser data sets. Without any sensor data synchronization procedures and with a laser driver acquiring at 5 frames per second, the laser acquisition was totally dismissed, and only the images were considered for evaluation benchmarking of the local ensemble detector in Chapter 4. The characteristics of these data sets are summarized in Table 3.2.

It is noteworthy that only with a recent inclusion of odometry and GPS modules, we were able to have information about speed and displacement of the vehicle.

Part II

The perception system

Chapter 4

Toward improving local detection by ensemble of classifiers

Contents

4.1 Classifiers **28**

4.1.1 MLP and SVM 28

4.1.2 Adaboost 29

4.2 Feature extractors **29**

4.2.1 Haar-like 30

4.2.2 HOG 32

4.2.3 LRF 34

4.3 Proposed ensembles **36**

4.3.1 HFI 36

4.3.2 HLSM-FINT 37

4.4 Experimental validation **41**

4.4.1 Parameter selection of the feature extractors 42

4.4.2 Evaluation of the component classifiers 42

4.4.3 Sensitivity analysis 43

4.4.4 Comparison of classifier fusion methods 45

4.4.5 Evaluation on NiSIS competition 48

4.4.6 Comparison of HLSM-FINT with other methods over Daimler-Chrysler data sets 49

4.4.7 Evaluation of HFI and HLSM-FINT over ISR-UC-imgs data set . . . 50

4.5 Conclusions **53**

If you just have a single problem to solve, then fine, go ahead and use a neural network. But if you want to do science and understand how to choose architectures, or how to go to a new problem, you have to understand what different architectures can and cannot do.
Marvin Minsky

A single feature extractor-classifier is not usually able to deal with the diversity of multiple image scenarios. Therefore, the integration of features and classifiers can bring benefits to cope with this problem, particularly when the parts are carefully chosen and synergistically combined.

Fusion of classifiers is a recent research area with the aim of improving individual classification performance. There are many ways of calling a fusion of classifiers: committee, ensemble, combination, integration, and so forth. Here, we adopted the name ensemble of classifiers.

The goal of this chapter is to present our two proposed ensemble detectors. For that, we firstly describe the main characteristics of the single feature extractors and classifiers we used in our proposed ensembles.

Feature extraction is an important processing stage in an object detection system, since it usually helps the classifier to separate the input space into object and non-object. Because of that, feature extractors received more attention in the development of our ensembles, and will be described in this chapter in more detail. For the classifiers, the main idea is given, while some mathematical background is properly described in the appendices when necessary.

4.1 Classifiers

4.1.1 MLP and SVM

SVM and MLP are both discriminant classifiers of the type $f(X) = sign(< W \cdot X > + \ b)$, where $X \in \mathbb{R}$ is the input vector, $W \in \mathbb{R}^N$ is a weight vector and $b \in \mathbb{R}$ is the bias component that adjusts the hyperplane $f(X)$ to better separate X. The main difference between these two classifiers is with respect to the way weights and biases are found. In SVMs, a quadratic and convex optimization problem is cast, and a kernel function is used for taking the input vector to a higher dimensional feature space, separating it linearly (refer to Appendix A, for more details).

SVM and MLP provide confidence scores, s_c, which are obtained by the Euclidean distance between the input vector and the separating hyperplane, $f(x)$. Many fusion methods (fuzzy systems and Bayesian networks, for instance) demand posterior probability or scaled confidences as inputs. To achieve a probabilistic output, it is necessary to scale the confidence scores by using a logistic link function [Platt 2000], which provides a probabilistic output, $P(s_c)$, such that

$$P(s_c) = \frac{1}{1 + exp(-\varrho_1 s_c + \varrho_2)}\,, \tag{4.1}$$

where ϱ_1 and ϱ_2 are usually obtained in the training process (in practice, ϱ_1 usually approaches to 1, while ϱ_2 approaches to 0); $s_c \in [0, \infty]$, and $P(s_c) \in [0, 1]$.

4.1.2 Adaboost

Adaboost is short for “adaptive boosting”. It is a classification system built on ensembles of weak classifiers developed by Freund and Schapire [Freund 1996]. These weak classifiers are trained in sequence (cascade), that is, each one is trained with weights extracted from the data, depending on the performance of the previous classifier. As a result, points of the input feature vector misclassified by one of the previous weak classifier are given greater weight when used to train the next classifier in the sequence. After that, the final classification is performed by combining all predictions through a weighted majority voting scheme. The final classification system is then formed by a rejection cascade. **Algorithm** 1 summarizes the main steps of adaboost classification.

For adaboost, the posterior probability, $P(Y = 1|x)$, is given by

$$P(Y = 1|x) = \frac{exp(2Y_m(x))}{1 + exp(2Y_m(x))} \tag{4.7}$$

4.2 Feature extractors

To facilitate the classification task, object images should have a representation which endows (hopefully) unique characteristics for each different object. This is the main role of the feature extractors, that is, provide high separability for the classification system.

The goal of this section is to describe the three types of feature extractors used in our ensemble detectors: Haar-like, HOG and LRF. The choice of these features relies on their characteristics of high degree of invariance to scale and lighting changes. However, it is noteworthy that Haar-like features are in general worser in small background-foreground

```
Initialize the weights, $\{w_n\}$, of the classifiers by setting $1/N$ for $n = 1, \ldots, N$.
for $m = 1$ to $M$ classifiers do
    1. Fit a classifier $y_m(x)$ to the training data by minimizing the weighted
    error function
        $J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)\,,$    (4.2)
    where $t_n \in \{-1, 1\}$ are binary target variables where 1 is object and $-1$ is
    non-object, and
        $I = \begin{cases} 1, & y_m(x_n) \neq t_n \\ 0, & \text{otherwise,} \end{cases}\,,$    (4.3)
    2. Evaluate the quantities
        $\varepsilon = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}\,,$    (4.4)
    and then use these to evaluate $\alpha_m = 1 - \varepsilon_m/\varepsilon_m$
    3. Update the weights
        $w_n^{(m+1)} = w_n^{(m)} exp\{\alpha_m I(y_m(x_n) \neq t_n)\}\,.$    (4.5)
end
Predict the final output, $Y_M(x)$, given by
    $Y_M(x) = sign\left( \sum_{m=1}^{M} \alpha_m y_m(x) \right)\,.$    (4.6)
```

Algorithm 1: Adaboost classification scheme

contrast, but they were kept here for comparison reasons. In Section 4.4, we experimentally evaluate these characteristics in a thorough analysis.

4.2.1 Haar-like

Haar-like features are types of square wavelets which were first applied densely by Papageorgiou and Poggio [Papageorgiou 2000] in pedestrian detection. Later, these features received improvements by Viola and Jones [Viola 2001] with respect to extraction time and addition of more prototypes (feature orientations). Next, we base our description of Haar-like features on Viola and Jones' implementation.

These features are represented by templates, which are comprised of a prototype, a

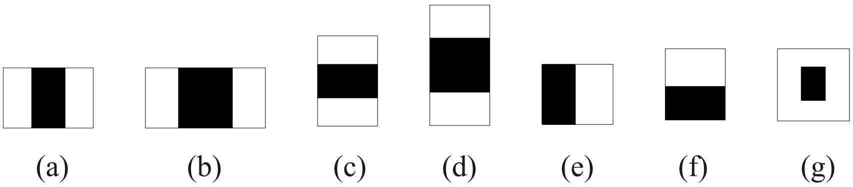


Figure 4.1: Haar-like features prototypes (edge orientations): (a), (b), (c), and (d) are line features; (e) and (f) are edge features, and (g) is the center-surrounded feature.

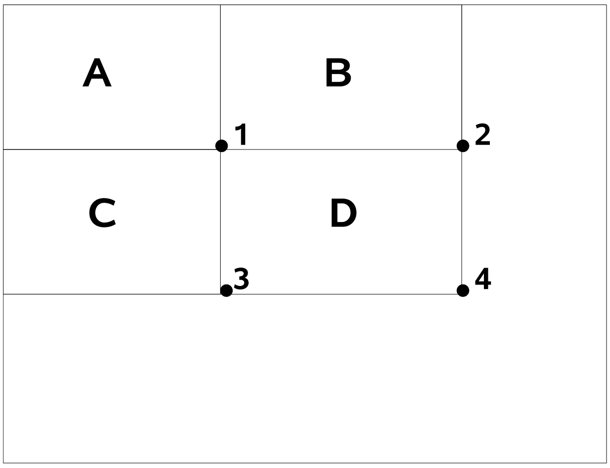


Figure 4.2: The sum of the pixels in rectangle D can be calculated with four references: 4 + 1 - (2 + 3).

size, and coordinates relative to the searching window origin. Figure 4.1 shows some types of edge orientations (prototypes), which are used to compute these features.

Haar-like features can be computed over an integral image to speed up the feature extraction step, by simply adding few reference points. An integral image, $ii(x, y)$, is achieved by the sum of gray levels of all pixels in the image, with the location (x, y) containing the sum of the pixels above and to the left, such that

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \,. \tag{4.8}$$

where $i(x, y)$ is the original image.

Figure 4.2 illustrates the way that a rectangle feature is obtained over an integral image.

The integral image is computed upon one pass over the original image, according to

```
input  : detWindow ← cropped gray-level image
output : featureVector ← Haar-like feature vector
shift sized prototypes (Fig. 4.1) onto horizontal and vertical directions with a
  certain stride;
for each prototype do
    featurePrototype ← extractFeatures();
    featureVector ← appendFeatures(featurePrototype);
end
normalizeFeatures(featureVector);
```

Algorithm 2: Haar-like feature extraction

$$s(x, y) = s(x, y - 1) + i(x, y)\,. \tag{4.9}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)\,. \tag{4.10}$$

where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$ and $ii(-1, y) = 0$.

Algorithm 2 illustrates how to compute these features over a detection window. Within the detection window, the overlapping prototypes of Fig. 2 are shifted with a certain stride, being computed according to Fig. 4.2, and finally appended to the feature vector. After that, the final feature vector is normalized by the L1-norm of the vector.

4.2.2 HOG

HOGs were used by Lowe [Lowe 1999] as descriptors over SIFTs, for object matching. Lately, they were modified by Dalal and Triggs [Dalal 2005] to be used densely with the aim of object detection. Two types of HOG features were analyzed in [Dalal 2005]: rectangular and log-polar. Dalal and Triggs demonstrated that the former outperformed the latter one. Hence, only the rectangular features are described here and applied in our ensemble detector. Figure 4.3 illustrates the way these features are extracted in a detection window.

HOGs are composed of blocks (set of cells), and cells (set of pixels), where, in turn, the oriented gradients are accumulated into bins. The size of the blocks and cells determines how fine the detection of the edge orientations is. **Algorithm** 3 describes how to compute these features.

First, orientation and magnitude of the edges are computed by using a centered mask [-1,0,1] onto horizontal and vertical directions, over gray-level images. Dalal and Triggs

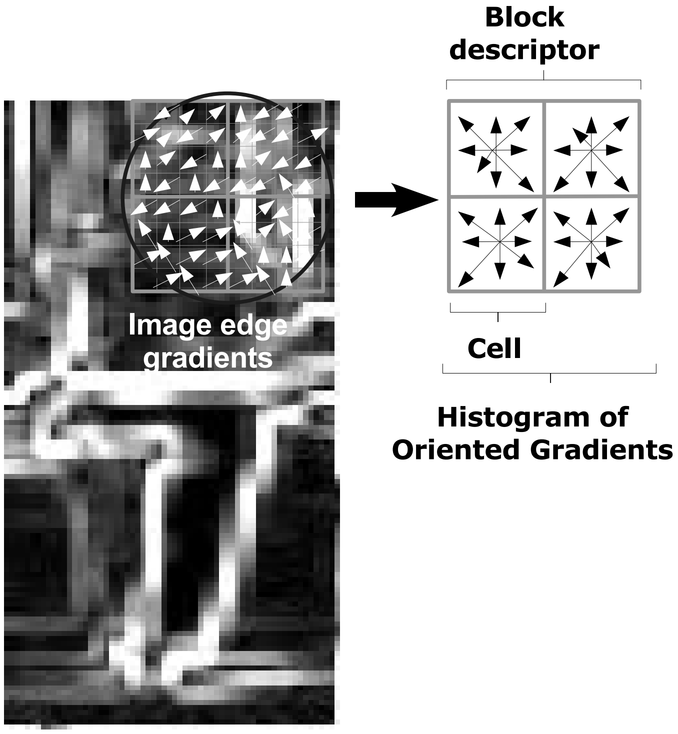


Figure 4.3: HOGs are computed densely within the detection windows, in the same way that the Haar-like features. The values of the gradients are accumulated into bins to form the histogram over the cells (right-most image).

```
input  : detWindow ←array of gradient angles and magnitudes
output: featureVector ←HOG feature vector
for each descriptor do
    for each block of the descriptor do
        for each cell of the block do
            gWeight ←gaussianWeight();
            cWeight ←cellWeight();
            featureVector ←curMag * gWeight * cWeight;
        end
    end
end
normalizeFeatures(featureVector);
```

Algorithm 3: HOG feature extraction

[Dalal 2005] showed that, in pedestrian detection, a gamma compression over RGB images ($\sqrt{RGB}$) presents better performance, although not significantly higher with respect to gray-level images. For each detection window, HOG descriptors are computed in a dense way (with overlapping descriptors), with horizontal and vertical strides. Histogram bins are accumulated over the magnitude of the gradients, after weighting its bin. For pedestrian detection, it was demonstrated that unsigned HOGs ($0 - 180^0$) present higher performance than signed ones ($0 - 360^0$). Each bin is weighted by a gaussian function (line 4), g_{weight}, such that

$$g_{weight} = \frac{exp\left(\frac{-dist^2}{\sigma^2}\right)}{\pi * \sigma}, \tag{4.11}$$

where σ is half of the descriptor width, $dist$ is the Euclidean distance between each bin and the center of the descriptor.

The cell weight (line 5), c_{weight}, is computed according to

$$c_{weight} = 1 - \min\left(\frac{dist}{width_{cell}}, 1\right), \tag{4.12}$$

where $width_{cell}$ is the cell width.

Finally, the feature vector is normalized (line 10). In [Dalal 2006], Dalal shows that different types of objects should have different types of normalization methods. For persons, L2-Hys (L2-norm with hysteresis) was used. This method is applied by computing L2-norm followed by clipping, and renormalizing, as in [Lowe 2004]. For all steps of HOG computation in pedestrian detection, several parameters may be used, and a set of default optimal values can be found in [Dalal 2005].

4.2.3 LRF

The receptive field (RF) of a neuron is a region in which neuron inputs cause neuron outputs to change behavior. Neuronal RFs can be categorized as local (LRF) or non-local, whether stimuli come from a bounded region or not [Haykin 2008]. If the neurons of an artificial neural network (ANN) have LRFs, then this ANN is an LRF neural network.

One successful example of an LRF neural network was proposed by LeCun and Bengio [LeCun 1995], based on a type of CNN (named as LeNet-5). It was applied on document recognition [Lecun 1998] and face recognition [Lawrence 1997], [Garcia 2004]. In a document recognition application, LeCun et al. [Lecun 1998] intuitively showed that this type of CNN with connections to specific regions (not fully connected) should demonstrate more stability, shift and scale invariance, and a decrease of output dimensionality. Later, however, Ridder et al. [Ridder 2003] experimentally found that a fully connected network might reach the same performance without loss of generality.

A CNN consists of parts, which characterize it as having some degree of shift, scale and distortion invariance. They are:

- **Feature maps** – composed of C and S layers, which are responsible for first convolving the input and then subsampling it, consequently reducing its dimensionality.

- **Weight sharing** – individual neurons of a feature map share a set of weights, decreasing the number of free parameters to be trained and allowing a parallel implementation of the neural network.

In Fig. 4.4, the architecture of CNN is depicted. The input layer of the CNN is a gray-level image with each pixel normalized to $(x-\mu)/\sigma$, where x is the gray value of the pixel, μ is the mean and σ is the standard deviation. Both μ and σ are calculated over all input pixels. This normalization has the advantage of speeding the convergence of the training process, also providing more invariance to illumination changes. The next four layers, C_1, S_1, C_2 and S_2, are in charge of extracting the features with different convolutional kernel masks. Each one of C_n layers is composed of feature maps, obtained by convolving rectangular regions of pixels (local receptive fields). S_n layers take the convolved regions of C_n layers, subsampling them with the goal of reducing dimensionality. The feature vector is obtained from the output of S_2. In the last 2 layers, an MLP neural network classifies S_2 features, backpropagating the error up to the C_1 layer's input, during the training stage. Hence, CNN is called a trainable feature extractor. Outputs $y_n^{(i,j)}$ of each C_n and S_n layers are given by:

$$y_n^{(i,j)} = b_n + \sum_{l=1}^{M}\sum_{s=1}^{K}\sum_{t=1}^{K} W_{n,l,s,t} X_l^{(\delta_w(i-1)+s,\delta_h(j+t))}\,, \tag{4.13}$$

$$y_n^{(i,j)} = b_n + \Upsilon \sum_{s=1}^{K}\sum_{t=1}^{K} X_k^{(\delta_w(i-1)+s,\delta_h(j+t))}\,, \tag{4.14}$$

with $n = 1...N$, where N is the number of feature maps; i and j are region coordinates of a feature map; X is the input vector; K denotes the size of a square kernel; M is the number of input images or feature maps; δ_w and δ_h are the strides (in pixels) between each application of the kernel onto width and height directions, respectively; $W_{n,l,s,t}$ represents the weight vectors of each output neuron; and Υ is a constant of subsampling.

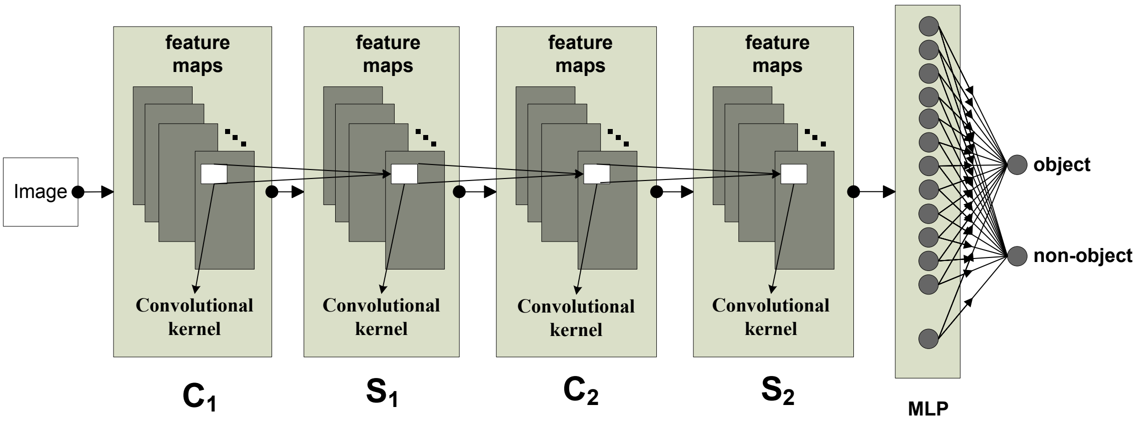


Figure 4.4: CNN architecture: The first 5 neural network layers correspond to the input gray-level image (input layer), C_1, S_1, C_2 and S_2, working as a trainable feature extractor. C layers convolve input image and the previous S layers, which, in turn, subsample C layers, reducing the dimensionality of the feature vector, obtaining only features which are hopefully important to the recognition process. In the last 2 layers, an MLP neural network is used to classify S_2 features. During the training, the error signal is backpropagated up to C_1 layer.

4.3 Proposed ensembles

Fusion of classifiers has been studied in the last few years with the aim of overcoming certain inabilities of single classification systems [Kuncheva 2004]. The objective is to explore diversity of the component classifiers in order to enhance the overall classification performance. In other words, since there is no perfect individual classifier yet, by assembling them, one can complement the other. If there are errors in the ensemble components, it is expected that they occur on different image objects in order to give rise to fusion methods that improve the performance of the whole system. Additionally, the rationale of building ensembles is also that most individual classifiers agree in a certain way, such that the whole system can be more successful than its component parts.

In this section, we describe our two proposed ensemble of classifiers: HFI and HLSM-FINT. The former is a generic approach based on a hierarchy of fuzzy systems; the latter is a classifier-driven approach with the aim of getting a synergistic architecture.

4.3.1 HFI

The intuition behind HFI can be summarized by two main ideas, as follows.

- Find an overlapping rate between the detection windows of the component classifiers;

- Evaluate the scaled confidence of each classifier.

The architecture of the proposed fusion system with two component classifiers as input is depicted in Fig. 4.5. The goal is not only to evaluate the overlapping rate of the detection windows, represented by the bounding boxes of each detected object, but also to weigh the score of each classifier in order to discard or to maintain the joint decisions. In the first stage, three types of variables have been used in parallel: the rate between perimeters of the detection windows, Euclidean distance between the centers of the detection windows divided by the highest window width, and the scaled confidences[1] of the two component classifiers, $C1$ and $C2$.

Although it is possible to add more classifiers, since HFI is composed of a cascade of fuzzy systems, the more classifiers are included, the more intractable would be the combination of fuzzy variables and rules to guarantee an increase in the fusion performance. Hence, we have limited the use of HFI on pairwise applications.

Note that if just one of the classifiers detected an image object, the window overlapping rate (intersection rate) approaches to zero, and only the joint confidence of this single classifier will be considered in the fuzzy inference. The parameters of the fuzzy systems are illustrated in Fig. 4.6. Fuzzy variables are represented with their fuzzy sets in Figs. 4.6(a)-4.6(f). The set of fuzzy rules over the fuzzy variables is represented as knowledge surfaces depicted in Figs. 4.6(g)-4.6(i).

4.3.2 HLSM-FINT

Rather than a generic fusion method as HFI, HLSM-FINT was built on carefully chosen components (feature extractors and classifiers). The goal is to structure an ensemble which presents a synergistic performance in various situations. Therefore, the architecture of HLSM-FINT has emerged from a experimental analysis with some other fusion methods and over different data sets.

The choice of the feature extractors, HOG and LRF, was motivated by the studies found in [Dalal 2005], [Munder 2006] and [Szarvas 2006]. Dalal and Triggs [Dalal 2005] presented an experimental analysis demonstrating that HOG features outperforms PCA-SIFT, Haar wavelets and shape contexts in a complex data set. Munder and Gavrila [Munder 2006] also experimentally showed that LRF features present superior performance in comparison with PCA and Haar wavelets, although computed from an MLP

[1] These scaled confidences are obtained by Eqs. 4.1 and 4.7. Since we are integrating classifiers in a fuzzy framework, rather than in a probabilistic framework, we will refer to these scores as “scaled scores” or “scaled confidences”, rather than posterior probabilities.

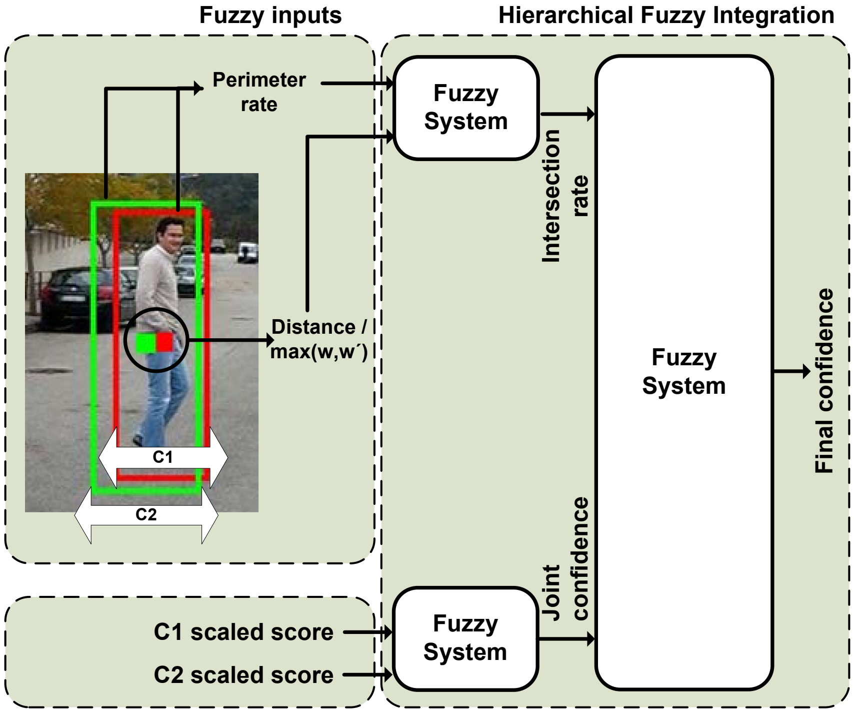
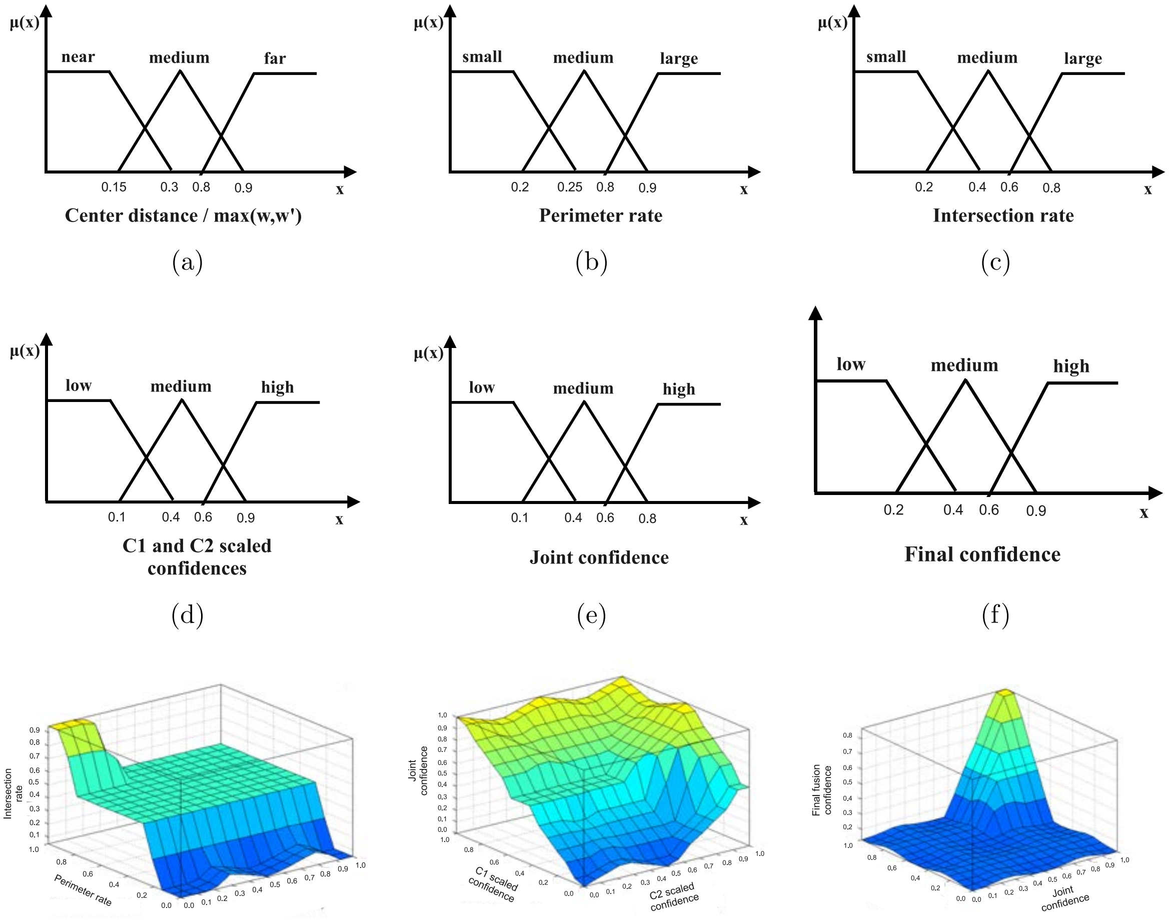


Figure 4.5: HFI architecture. Three variables are considered in the first layer of the fusion system: rate between the perimeters of the detected windows, Euclidean distance between the two centers of the detection windows divided by the highest detection window width, and the scaled scores of two classifiers; in the second layer, the output for each pair of fuzzy variable feeds another fuzzy system in order to give the final confidence score. Although it is possible to include more classifiers, it would be difficult to manage the exponential growing of the cascade fuzzy rules.

over Haar wavelets, or PCA features, and classified by SVM or Adaboost, which turned it to a method more sensitive to lighting variations. Szarvas et al. [Szarvas 2006] found that LRFs built on CNNs have great potential in pedestrian recognition. Our goal is thus to show that there is an opportunity to integrate synergistically the outputs of high performance classifiers performing over these two types of features.

Since we are working on specific feature extractors and classifiers, it is necessary to find the best combination of them in order to ensure the best performance. Let us consider the initial ensemble (combination) of classifiers depicted in Fig. 4.7. The initial ensemble is structured by HOG and LRF features, with each feature vector being classified by an MLP and an SVM. In this stage, the classifier fusion method is irrelevant, because we are focused on which classifiers to include in the ensemble. Henceforth, consider the following notation for each pair of extractor-classifier: HS = HOG/SVM, HM = HOG/MLP, LS = LRF/SVM, LM = LRF/MLP.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.6: Fuzzy variables and knowledge surfaces: (a), (b) and (d) represent the four fuzzy input variables of the first stage; (c) and (e) are fuzzy output variables of the first stage, which feed the fuzzy system of the second stage; (f) is the fuzzy output variable of the second stage, giving the final fusion confidence. (g), (h) and (i) are the resulting knowledge surfaces of the application of the fuzzy rules over the fuzzy variables.

4.3.2.1 Refining the Initial Ensemble

There is much discussion whether ensembles of less accurate classifiers can outperform ensembles of more accurate classifiers with less diversity [Kuncheva 2004], [Zenobi 2001]. Here, we found that ensembles of accurate classifiers have the following advantages:

- More stable consensus among the parts.
- In presence of lighting variation, where the single classifier performance is naturally dropped, the consensus and diversity of opinions can bring higher benefits to the

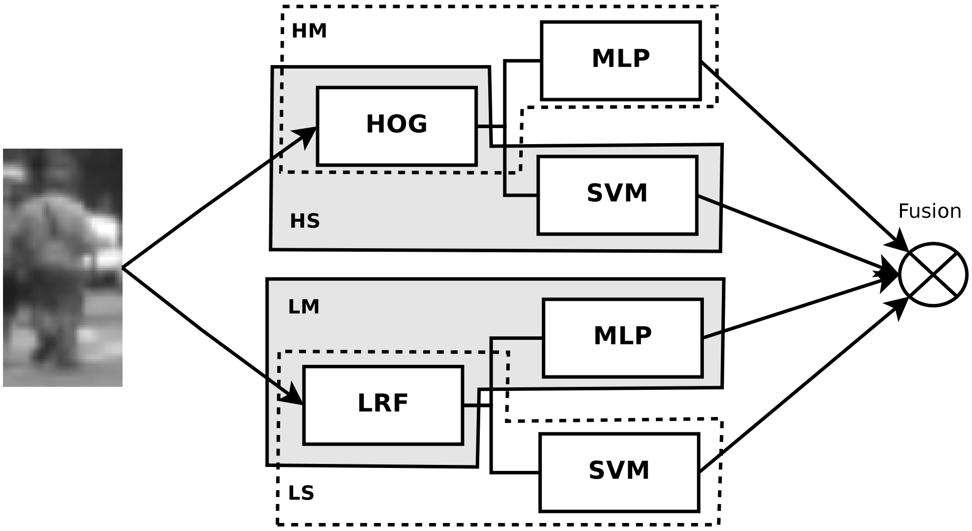


Figure 4.7: Initial ensemble. Composed of two feature extractors, HOG and LRF (provided by the CNN), and four classifiers, two MLPs and two SVMs. Abbreviations: HS = HOG/SVM, HM = HOG/MLP, LS = LRF/SVM, LM = LRF/MLP

ensemble decision.

Rather than simply relying on the accuracies of individual classifiers to choose the best combination of feature extractors and classifiers, we are interested in a causal relationship between diversity and ensemble accuracy as experimentally demonstrated in [Aksela 2006]. The initial ensemble architecture depicted in Fig. 4.7 has been used as a basis for a diversity analysis in order to determine the best combination of classifiers to integrate the final ensemble. In this regard, we have considered the exponential error count proposed by [Aksela 2006] to refine the ensemble. This diversity measure counts the importance of the component classifiers not making the same errors too often, also considering the correct classification by scaling the diversity measure. This assumption is very beneficial to our approach since we are working on component classifiers with high accuracy. The best set of classifiers is found by taking the one with the lowest EEC, E_{ec}, given by

$$E_{ec} = \frac{\sum\limits_{i=1}^{K} (N_{same}^{i \times 0})^i}{N^{K \times 1} + 1}\,, \tag{4.15}$$

where K is the number of classifiers in a set, $N_{same}^{i\times 0}$ denotes the count of errors made by a total of i classifiers to the same class, and $N^{K\times 1}$ denotes the number of testing samples for which all classifiers in the set are correct.

Table 4.1: Exponential error count (E_{ec}) for classifier combinations from the initial ensemble

Combination of classifiers	E_{ec}
HS + LM + LS	5.01
HS + HM + LS	9.42
HM + LM + LS	13.07
HM + HS + LM + LS (initial ensemble)	15.89

The rationale of this diversity measure is thus of counting the frequency of the same mistakes made by groups of classifiers in a set, scaling the result by the total agreement within the set. It is noteworthy that, by doing this, one relates not only the accuracy of the classifier but also the coincident error of the set of classifiers. The E_{ec} for each combination of classifiers from the initial ensemble was computed by using the DaimlerChrysler training data sets (see Table 2.1). Each classifier was trained by using the three training data sets, consequently obtaining three classification models. After that, each one of those models were used to classify the two validation data sets. The final result was then averaged by those six combinations, as in [Munder 2006].

The values of E_{ec} for combinations of the classifiers within the initial ensemble (Fig. 4.7) are summarized in Table 4.1. The set of classifiers HS + LM + LS presents the lowest E_{ec}. When the initial ensemble is considered, then the value of E_{ec} is the highest one. Pairwise classifiers over the initial set of classifiers (Fig. 4.7) had usually higher E_{ec} values than those presented in the table, because they tend to fail at the same time too often.

4.4 Experimental validation

In this section, an experimental analysis is described to assess the performance of the proposed fusion architectures: HFI and HLSM-FINT. The evaluation methodology consisted of two experiments over crop-wise DaimlerChrysler data sets (see Table 2.1), and over a video sequence (#Seq. 3 in “ISR-UC-imgs” data sets; see Table 3.2).

In the first evaluation round, our goal was to assess the performance of HLSM-FINT over the DaimlerChrysler data sets and a lighting transformed version of these data sets. These latter ones were created by applying two artificial lighting transformations on the DaimlerChrysler validation data sets. In this step, a comparison with other fusion methods was also included. Since HFI uses the geometry of the detection window as a fuzzy input variable, it is not possible to evaluate this fusion method over cropped images (in this case, only the scaled scores of the classifiers would be considered). This led us to a second evaluation round, considering the best HLSM-FINT parameters, found in the first round. The classification performance was analyzed by means of ROC curves.

4.4.1 Parameter selection of the feature extractors

In this section, the best parameters of the component feature extractors and classifiers used in the HLSM-FINT were obtained by using DaimlerChrysler validation data sets.

We implemented a variant of LeCun and Bengio's CNN [LeCun 1995], with full connections and just one hidden layer in the MLP (see Fig. 4.4 for the layout of the architecture). The number of C and S feature maps and the number of hidden neurons in the MLP were varied on a cross validation procedure. The size of the kernels has been kept at the finest possible resolution as in [Lecun 1998]: $C_1 = 5 \times 5$, $S_1 = 4 \times 4$, $C_2 = 2 \times 2$ and $S_2 = 3 \times 3$.

As shown in Fig. 4.8, the best LRF parameters were found to be $F_{C_1} = 4$, $F_{S_1} = 4$, $F_{C_2} = 14$, $F_{S_2} = 14$ and $HN = 25$, where F_{C_n} and F_{S_n} represent the numbers of feature maps in C and S layers, respectively, and HN is the number of hidden neurons in the MLP layer. With these parameter values, the LRF network reached a 94% hit rate (HR). The HRs in Fig. 4.8 were found by considering the highest false alarm rate (FAR) in the curve; in other words, the thresholds of the classifiers scores were kept to be zero in the cross-validation procedure.

For HOG parameters, we used Dalal's work as reference [Dalal 2006]. Edges were computed by a centered surrounded convolution mask [-1, 0, 1], onto vertical and horizontal directions; the gradients of the edges were calculated in regions of 3x3 pixel cells and 2x2 cell blocks (block descriptors are then 6x6 pixel wide), with a block stride of 2 pixels, preventing falling off the boundaries. The histogram of the edges was calculated in a half circle, composed of 9 bins. After extracting the features, the feature vector was normalized by applying an L2-Hys norm.

4.4.2 Evaluation of the component classifiers

All curves plotted over DaimlerChrysler data sets were obtained by training the classifiers with the three DaimlerChrysler training data sets, averaging the result of the classification models over the two DaimlerChrysler validation data sets, following the same methodology as in [Munder 2006].

Figures 4.9(a) and 4.9(b) show the ROCs of the individual classifiers. For LM, the best parameters shown in Fig. 4.8 were used. For LS and HS, three types of SVM kernels –

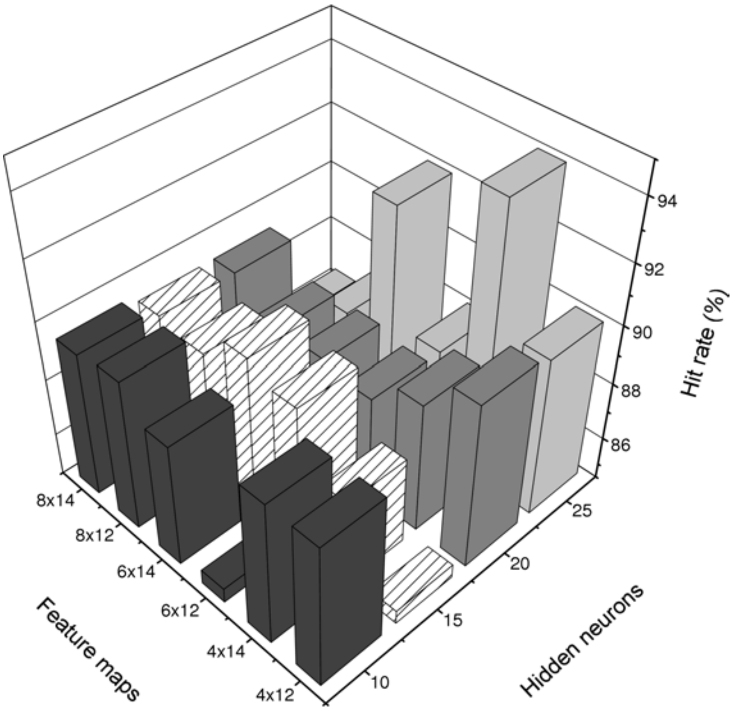
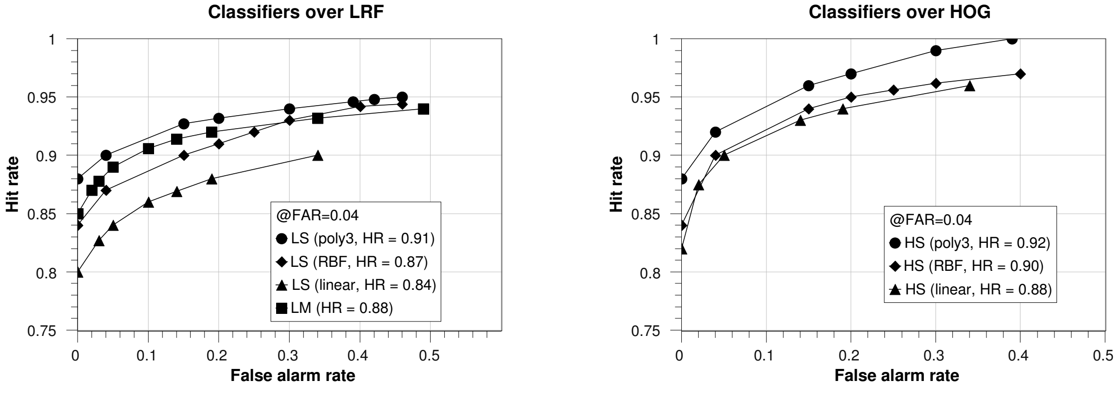


Figure 4.8: LRF parameters: Parameters were found by varying the number of C and S feature maps, and the number of hidden neurons. The parameter combination with the best performance was $F_{C_1} = 4$, $F_{S_1} = 4$, $F_{C_2} = 14$, $F_{S_2} = 14$ and $HN = 25$, with a hit rate equal to 94%.

linear, third degree polynomial (poly3) and radial basis function (RBF) – were evaluated: The poly3 kernel presented the best performance for both SVM classifiers over LRF and HOG; the best points in the ROCs were chosen to be at 4% of FAR, where LS-poly3, LM and HS-poly3 reached 91%, 88% and 92% of HR, respectively.

4.4.3 Sensitivity analysis

Two artificial lighting transformations have been applied to modify the DaimlerChrysler validation data sets with the aim of creating the effect of shadowy and sunny (overexposure to sunlight) effects. The shadowy effect was obtained by applying $I'_{shad}(x,y) = I(x,y) - \frac{2\delta}{w-1}y + \delta$, where $I(x,y)$ is the original image of w width, and δ is a pixel constant equal to 80 in our experiments; the sunny effect was obtained by applying a multi-scale retinex (MSR) algorithm [Rahman 1996]. MSR was originally designed to image enhancement by estimating scene reflectance from the ratios of scene intensities. In our experiments, MSR parameters were taken to produce brighter images, simulating a sunlit effect on the objects in the scene.



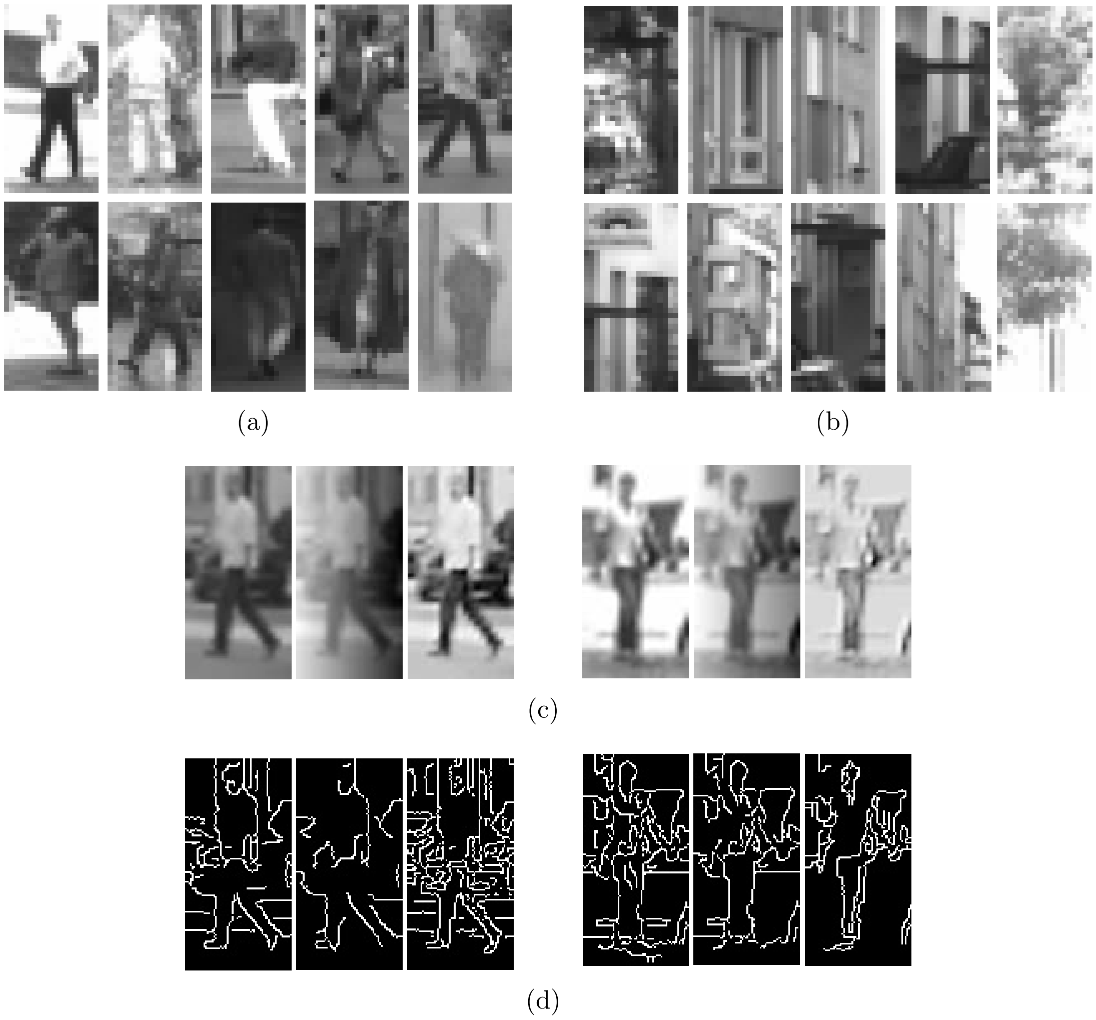
(a) (b)

Figure 4.9: ROCs of the component classifiers: Curves show performance of the individual classifiers over DaimlerChrysler data sets. In (a), performance of MLP, linear SVM, RBF SVM and poly3 SVM over LRF features are depicted; in (b), SVM with linear, RBF and poly3 kernels are plotted over HOG features.

Figure 4.10(c) depicts some image samples of the artificial transformations applied to DaimlerChrysler validation data sets. Figure 4.10(d) shows, in turn, the effect of the lighting transformation on the edge information of each image; it can be observed that the shadowy effect makes the image to lose some edge information since this transformation obscures part of the image. On the other hand, the sunny effect depends on the illumination of the image: If the image is under illuminated (original image on the left in Fig. 4.10(c)), the dark areas increase the contour of the image, causing more edges to appear, whereas application on a brighter image (original image on the right in Fig. 4.10(c)) leads the image to have brighter areas and consequently to a loss of some edge information. This side effect of the light transformations mainly influences the individual classifiers, since they are based on edge detection, causing a decrease of their performances. This fact also produces an increase of the diversity of the classifiers (as the agreement among them decreased), and the fusion method can explore this synergism to raise the overall performance of the system.

The ROC curves of the individual classifiers, after the lighting transformations, can be seen in Figures 4.11(a) and 4.11(b), where the points highlighted in the boxes correspond to the same scores of the classifiers at 4% of FAR in the ROC curves of Figures 4.9(a) and 4.9(b).

Table 4.2 summarizes the results of the best individual classifiers achieved in the ROC curves of Fig. 4.11. Note that whereas LS and LM decreased by approximately 3 and 2 percentage points of HR, HS decreased by 2 percentage points, under shadowy effect;


(a)

(b)

(c)

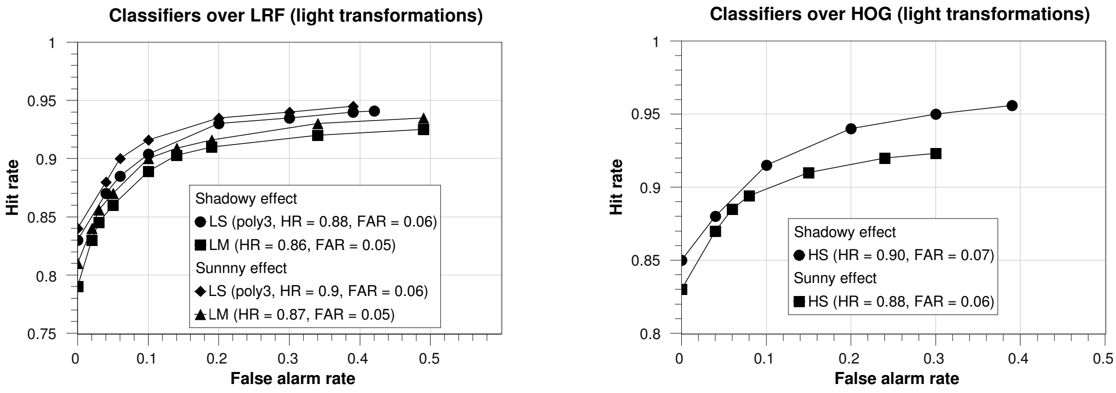
(d)

Figure 4.10: Some samples of DaimlerChrysler data sets (a) and (b), and lighting transformed version (c): From left to right, original image, shadowy and sunny effects. In (d), edge information after the lighting transformation over (c).

under sunny effect, LS and LM decreased by 1, and HS decreased by 4 percentage points of HR, respectively. These individual classifier behaviors demonstrate that they were affected oppositely with respect to those two lighting transformations, giving rise for the fusion method to balance the overall performance.

4.4.4 Comparison of classifier fusion methods

In this section, we evaluate a set of fusion methods over our feature extractors-classifiers (refer to Appendix B for a description of these methods). The compared fusion methods are: fuzzy integral (FI) (Appendix B.2) and majority vote (MV) (Appendix B.1). In the



(a) (b)

Figure 4.11: Evaluation of lighting transformation applied to DaimlerChrysler validation data sets: (a) and (b) illustrate the performance of the best individual classifiers.

Table 4.2: Comparative results after lighting transformations

	Original data set	**Sunny effect**	**Shadowy effect**
Classifier	**HR/FAR (%)**	**HR/FAR (%)**	**HR/FAR (%)**
LS (poly3)	91.0/4.0	90.0/6.0	88.0/6.0
LM	88.0/4.0	87.0/5.0	86.0/5.0
HS (poly3)	92.0/4.0	88.0/6.0	90.0/7.0

former, Sugeno [Sugeno 1974] and Choquet [Choquet 1954] types were used. In the latter, sum, weight and heuristic rules were applied. The heuristic rule is a mixed of sum and weight rules, conceived to strengthen the characteristics of HOG, better to detect non-pedestrians, and LRF, better to detect pedestrians (according experimental evidences).

Before presenting the results of each fusion method, some notes on the parameters of fuzzy integral have to be done. Let $N_c = 3$ be the number of classifier outputs, then eight fuzzy measures, $g(.)$, must be defined. The initial values for $g(P(s_{c1}))$, $g(P(s_{c2}))$ and $g(P(s_{c3}))$ have been chosen to be: 0.15, 0.24 and 0.30, for LM, LS and HS, respectively, which come from the best points in the ROC curves of the scaled scores, $P(s_{cn})$, of the individual classifiers, given by (4.1). The fuzzy measures of the other aggregated subsets, g({P(s_{c1}),P(s_{c2})}), g({P(s_{c1}),P(s_{c3})}), g({P(s_{c2}),P(s_{c3})}), can now be calculated from (B.4) using $1 + \lambda = (1 + 0.15\lambda)(1 + 0.24\lambda)(1 + 0.30\lambda)$. After finding the fuzzy measures, a threshold was chosen to be the value of the minimum fuzzy measure of a set of two classifiers. In other words, the threshold for the FIs was g({P(s_{c1}),P(s_{c2})}) ≥ 0.47. This value indicates that one should rely on a fuzzy output, if it is greater than the fuzzy measure of the set formed of LM and LS.

Table 4.3: Results of the fusion methods over DaimlerChrysler data sets

Fusion Method	**HR/FAR (%)**
Heuristic MV	**96.5/1.9**
Sum MV	94.2/3.9
Weighted MV	94.8/3.1
Sugeno FI	**96.4/2.4**
Choquet FI	95.8/2.5

Table 4.4: Result of the fusion methods over transformed DaimlerChrysler data sets

	Sunny effect	**Shadowy effect**
Fusion Method	**HR/FAR (%)**	**HR/FAR (%)**
Heuristic MV	**94.9/3.2**	**93.1/3.1**
Sum MV	92.3/4.6	91.7/4.2
Weighted MV	92.5/4.8	91.6/4.9
Sugeno FI	**94.7/3.4**	**92.9/3.3**
Choquet FI	92.7/4.1	91.7/3.9

Table 4.3 presents the results of the fusion methods over DaimlerChrysler data sets. It is worth noting that Sugeno FI and the heuristic MV present similar performance, although the use of Sugeno FI can provide a more comprehensive framework and theoretical basis for our purposes.

The effect of the lighting transformations in the ensemble is summarized in Table 4.4. Interestingly, despite a decrease in the performance of the individual classifiers caused by the lighting transformations, the classifier fusion balanced those losses by combining different characteristics of the component classifiers.

Table 4.5 summarizes the global performance results, considering the best fusion methods compared with the component classifiers over DaimlerChrysler and lighting transformed DaimlerChrysler data sets. Over DaimlerChrysler data sets, the best fusion methods (heuristic MV and Sugeno FI) increased by approximately 8.5, 5.5 and 4.5 percentage points of HR from the best LM, LS and HS, respectively; while there was a decrease by 2.1 and 1.6 percentage points in FAR for heuristic MV and Sugeno FI, respectively. The last line on the table presents the average performance over all data sets. Heuristic MV and Sugeno FI methods present similar average performances, with a difference of 0.2 percentage points of HR and 0.3 of FAR between them. It is worth noting an average gain of approximately 4.7 percentage points from HS (the best individual classifier) in comparison with heuristic MV and Sugeno FI fusion methods. Furthermore, by comparing fusions after the lighting transformations, in the worst case (Sugeno FI, DaimlerChrysler shad-

Table 4.5: Summary of the results over DaimlerChrysler and transformed DaimlerChrysler data sets (HR/FAR (%))

	Fusion		**Individual classifiers**		
Dataset	**Heuristic**	**Sugeno**	**LS**	**LM**	**HS**
DaimlerChrysler	96.5/1.9	96.4/2.4	91/4	88/4	92/4
DaimlerChrysler (sunny)	94.9/3.2	94.7/3.4	90/6	87/5	88/6
DaimlerChrysler (shadowy)	93.1/3.1	92.9/3.3	88/6	86/5	90/7
Average	**94.8/2.7**	**94.6/3.0**	**89.7/5.3**	**87/4.7**	**90/5.7**

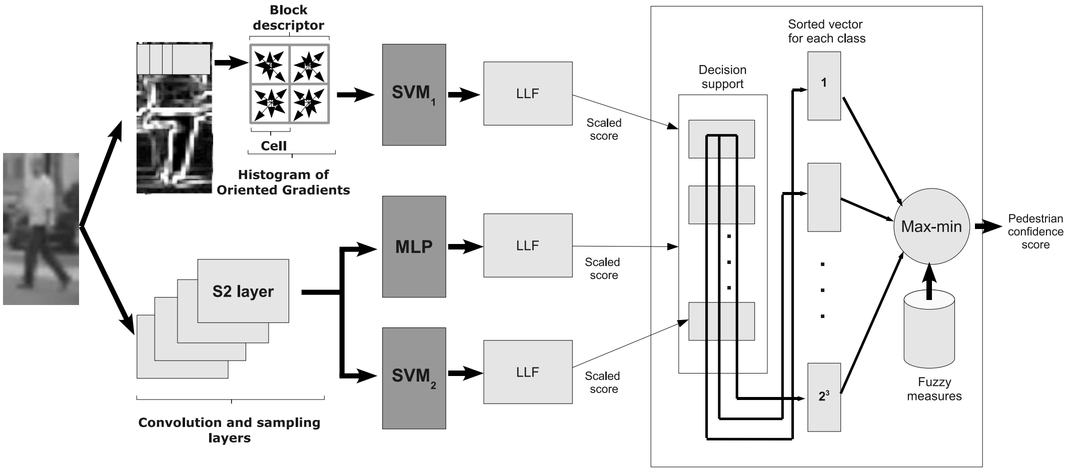


Figure 4.12: Final ensemble architecture: The component classifiers provide scores with values $[0, +\infty]$, which are scaled to $[0, 1]$ by a logistic link function (LLF). Finally, the Sugeno FI combines the classifier results, providing another confidence score $[0, 1]$, which is thresholded by the minimum value of the combination of two classifiers.

owy), the performance of the ensemble decreased by approximately 3.5 percentage points of HR, while increased by 0.9 percentage points of FAR with respect to the classification fusion without the lighting transformations.

According to the results, the final architecture of our proposed method is then illustrated in Fig. 4.12. This was called HLSM-FINT.

4.4.5 Evaluation on NiSIS competition

The proposed ensemble method with the heuristic MV (instead of sugeno FI) won the most accurate model award in 2007 Nature-inspired Smart Information Systems (NiSIS) international competition, out of 16 participants [Oliveira 2007b]. The model was eval-

Table 4.6: Comparison over DaimlerChrysler data sets

Classifier method	**HR/FAR (%)**
HS (poly3)	92.0/4.0
LS (poly3)	91.0/4.0
LM	88.0/4.0
HLSM-FINT	96.4/2.4
SVM on LRF (bootstrapped) [Munder 2006]	90.0/5.0
Sum MV on GLB+LEM (non-bootstrapped) [Nanni 2008]	90.0/5.0

uated on a subset[2] of DaimlerChrysler data sets, containing 1225 training images, 2450 labelled images, for validation of the algorithms, and 6125 unlabeled images, for measuring the performance in the competition. Images on testing and validation data sets were artificially occluded. In the pedestrian classification challenge, our proposed method achieved a classification accuracy of 95.97%.

4.4.6 Comparison of HLSM-FINT with other methods over DaimlerChrysler data sets

Table 4.6 summarizes the HR/FAR of our component and ensemble classifiers, as well as the results of two other methods: SVM on LRF (bootstrapped) [Munder 2006], and Sum MV on GLB+LEM (non-bootstrapped) [Nanni 2008]. As can be noticed, even the individual classifiers LS and HS outperform those two methods concerning HR and FAR. HLSM-FINT outperforms the methods in [Munder 2006], [Nanni 2008] in 6.4 percentage points of HR, and in 3.1 percentage points in FAR. It is worth noting that:

- The final result in [Munder 2006] is achieved after an increment in the number of training images (bootstrap), while this additional step has not been applied neither to our work nor to [Nanni 2008].

- As mentioned before, the generation of LRF features in [Munder 2006] involved the training of an MLP applied to Haar-like features and PCAs; while, in our work, the LRFs obtained from a CNN led to more invariance on illumination change and image shifts.

[2] These data sets are provided in http://www.isr.uc.pt/∼lreboucas "as they were" since they are no longer publicly available on the NiSIS website.

4.4.7 Evaluation of HFI and HLSM-FINT over ISR-UC-imgs data set

In addition to the study and conclusions presented over DaimlerChrysler and transformed DaimlerChrysler crop-wise data sets, HLSM-FINT has been tested on a video sequence gathered at the engineering campus of Coimbra University. This video sequence has 364 frames (640x480 pixel frames resized to 320x240) with 430 annotated pedestrians (ground truth) in different poses and interactions (#Seq. 3 of Table 3.2).

In order to locate and recognize the pedestrians in each frame, a sliding window technique has been applied with the goal of analyzing the trade-off between recognition performance and speed. After finding the detected windows, they were clustered by a non-maxima suppression algorithm[3]. Only pedestrians at a distance up to 25 meters were considered for annotation, as constrained by a laserscanner used to gather the data set. A pedestrian was successfully matched if an overlap criterion was met

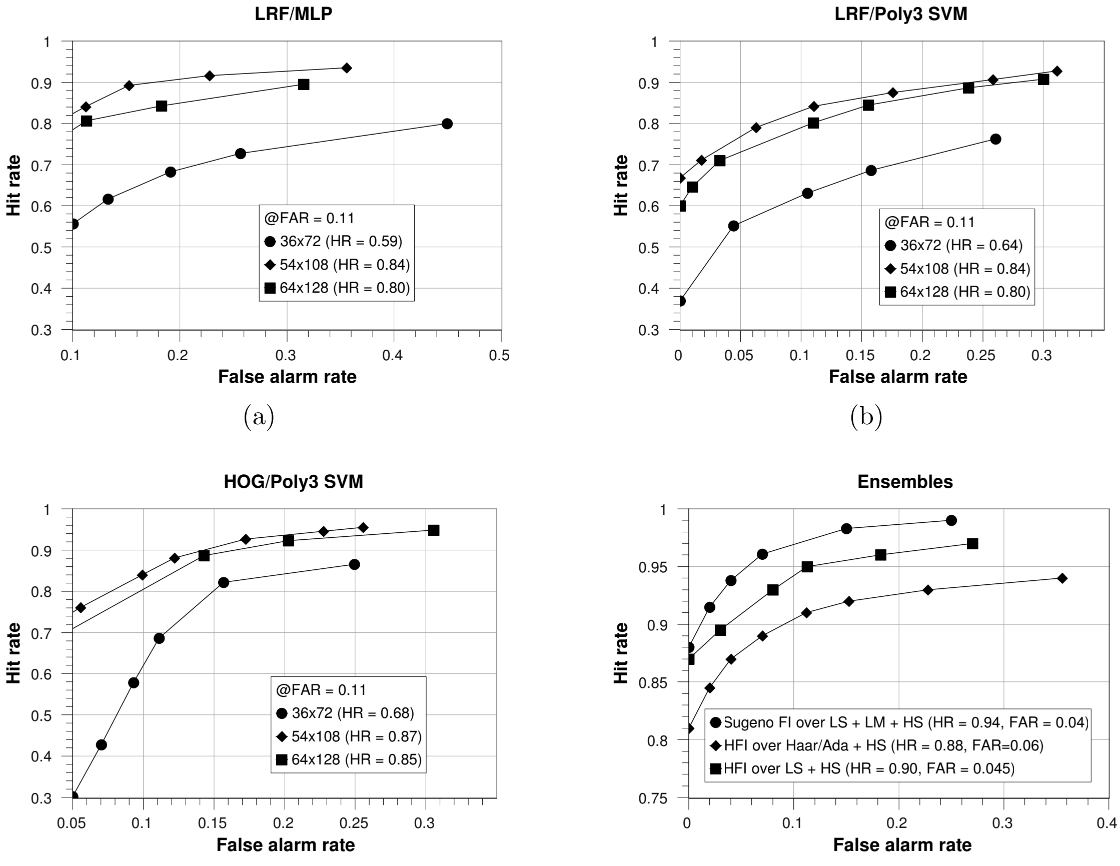
$$A_{Overlap} = \frac{A_{gt} \cup A_{det}}{A_{gt} \cap A_{det}} \geq 0.4\,, \tag{4.16}$$

where A_{gt} corresponds to the area of the ground truth bounding box, and A_{det} is the area of the detected bounding box. If $A_{Overlap}$ is greater than or equal to 40% then the detected bounding box is considered a hit.

Since a 18x36 pixel window with three strong classifiers is not a suitable candidate for a proper recognition speed, we have decided to double and triple the size of the cropped images, considering yet a window size of 64x128 pixels [Papageorgiou 2000], [Dalal 2005], making the stride of the searching window equal to 1/8 for the 64x128 window size and 1/9 for the rest, preventing it falling off boundaries. Doubling and tripling the DaimlerChrysler images would introduce enough distortion, then the choice was to use another data set to guarantee more stability in the training stage. This way, the INRIA person data set [Dalal 2005] was chosen (see Table 2.2), resizing the 64x128 pixel images to 36x72 and 54x108 pixel wide. This data set has the advantage of presenting a fixed height for the pedestrians in the image, these being centered on a border of 16 pixels on each side. These characteristics provide a more stable searching in a sliding window procedure [Dalal 2005], [Papageorgiou 2000].

Figures 4.13(a)-4.13(c) depict the ROC curves of the video sequence evaluation. For the three feature extractor-classifiers, we found that a 54x108 pixel window achieved the best performance. By adjusting the thresholds of the classifiers to lie on 11%, the

[3] This algorithm will be properly described in Chapter 6.



(a) (b)

(c) (d)

Figure 4.13: ROCs of the video sequence evaluation: (a), (b) and (c) show the evaluation of different sliding window sizes, considering also strides of 1/8 for the 64x128 pixel window and 1/9 for the others. In (d), ROCs of the evaluation among three types of ensemble: proposed ensemble using Sugeno FI as a fusion method, HFI over Haar wavelet/adaboost + HS, and HFI over LS + HS.

HRs of 84%, 84% and 87% were obtained by LM, LS and HS, respectively. 36x72 pixel windows showed generally more false positives per frame (considering all frames in the video sequence). For HS, 54x108 and 64x128 pixel windows presented similar performance with a small difference of 2 percentage points. The best parameters found in the crop-wise analysis have been increased proportionally as the windows increased. In this regard, CNN had the kernel sizes doubled or tripled, according to the increase of the window (for the 64x128 pixel window, the values were inherited from the 54x108 pixel window). The number of C_1 and S_1 feature maps have not changed, although the size of the kernels in each layer was reduced to the half in order to keep the number of free parameters in the network under control. Concerning HOG parameters, for a 36x72 pixel window, a

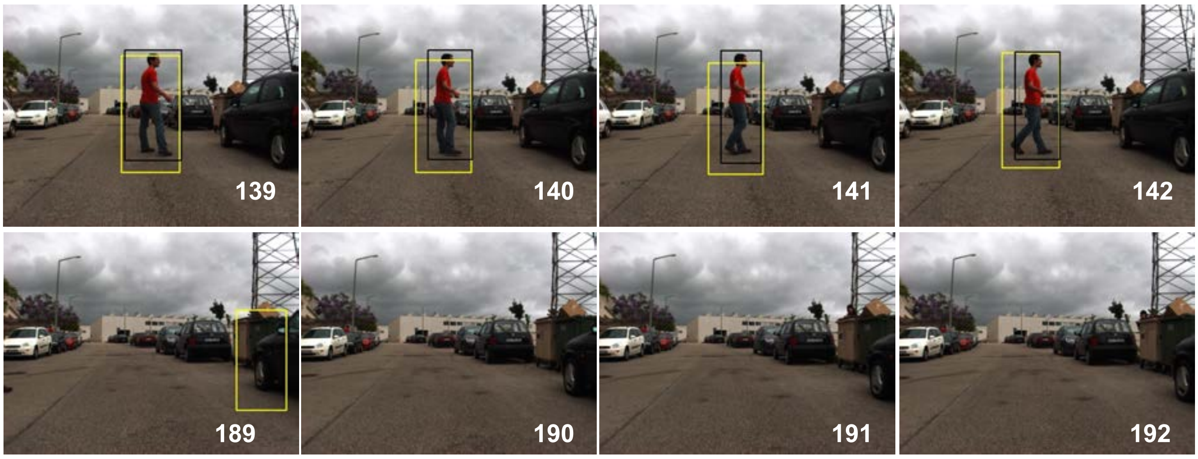


Figure 4.14: Examples of frame sequences: Annotations (ground truth) are the dark bounding boxes, while the light bounding boxes are the objects detected by the ensemble classifier. In frame 189, there is an example of a false alarm.

descriptor of 12x12 pixels with 2x2 cell block was applied, while, for 54x108 and 64x128 pixel windows, the best parameters provided in [Dalal 2006] were used, i.e., a 16x16 pixel descriptor, with 2x2 cell block.

In Fig. 4.14, two subsets of sequences are shown. The annotated objects are represented by the dark bounding boxes, while the detected (light) bounding boxes represent the result of the recognition by the proposed ensemble. Frame 189 shows an example of false alarm and no miss detection was encountered in those frame sequences.

4.4.7.1 Comparison with HFI

HFI was conceived to be applied on independent executions of classifiers, that is, there is no need to define the same window size or sliding window parameters as in HLSM-FINT. In this way, HFI has been used with two different pairs of classifiers: Haar wavelet/Adaboost + HS (originally used in [Oliveira 2007a]) and LS + HS. For Adaboost, a 24-stage cascade of classifiers presented the best performance. It is worth noting that as the training data set has been changed and the classifiers have been retrained, fuzzy measures of Sugeno FI had to be recalculated. The fuzzy measures obtained for LM, LS and HS were 0.18, 0.21 and 0.27, respectively, and the threshold of the fusion output was 0.46.

Considering the same individual classifier thresholds to represent their ROCs (see Figs. 4.13(a)-4.13(c)), it can be noticed in Fig. 4.13(d) that the proposed ensemble method outperforms the other two HFI based methods, with 94% of HR, reducing the FAR of 7 percentage points, while increasing the HR by 7 percentage points in regard to the best individual classifier (HS poly3).

4.5 Conclusions

The main goal of an image classification system is to recognize successfully the target object. By building single feature extractor-classifiers, one can often face problems, for instance with lighting changes, since finding the best trade-off between increasing the training data and classification performance is not a simple task. On exploring ensemble methods, one is able to create synergistic approaches in order to compensate the individual inability of the component classifiers in certain circumstances.

In this chapter, we presented two proposed ensemble methods: HFI and HLSM-FINT. This latter presented the best performance in a thorough experimental evaluation over several data sets, also in comparison with other external methods. Although presenting a state-of-the-art performance, HLSM-FINT was not evaluated over a data set with occlusion situations. This analysis will be done in Chapter 6 using another collected data set, collected to that end, wherein HLSM-FINT is used in a parts-based approach.

Chapter 5

Laser segmentation and labeling: A clustering-based approach

Contents

5.1 Outline of the laser-based detection system **56**

5.2 From laser points to labels **56**

5.2.1 Segmenting points 57

5.2.2 Labeling segments 60

5.3 Dealing with occlusion by spatial relationship between laser segments . . . **65**

5.4 Conclusions **66**

The world changed from having the determinism of a clock to having the contingency of a pinball machine.
Heinz R. Pagels, in "The Cosmic Code"

So far, we have based our pedestrian detection system on monocular vision. Although a sequence of monocular images offers highly dense data, from which a lot of information can be extracted, recovering depth information in images is always cumbersome. As it has been already discussed in Chapter 1, for a real-life pedestrian detection application, the use of multiple sources of information is crucial to provide a system with redundancy or complementarity. A possible way to tackle this problem is to employ more than one camera, which naturally brings additional efforts to guarantee object registration among the multiple images, acquired simultaneously.

Alternatively, integration of laser and monocular vision is still not fully explored. Object registration between these two sensors is straightforward after proper calibration, and depth information is always directly provided. The challenge is thus to increase the robustness of the integration and, consequently, of object detection.

In this chapter, we focus on laser object detection, as a way to provide a redundant and complementary object detection along with our image detector. The goal is then to have a laser pedestrian detection able to deal with partial segmentation, properly embodying contextual information in order to tackle difficult situations.

5.1 Outline of the laser-based detection system

Our laser-based detection system follows a coarse-to-fine strategy in order to provide a proper laser-point segmentation of the pedestrians in the scene. After labeling the fine segments (within coarse segments), a Procrustes analysis is performed with the aim of assigning a confidence for each fine segment as parts of human body (i.e., arms and torsos). Since the process of segmenting the laser points is a complex task, it is necessary to cope with many issues in order to address the segmentation and labeling processes coherently. This way, we make the final decision of naming a segment as a pedestrian or non-pedestrian, by contextually analyzing the spatial relationship among this segment and its adjacent segments.

In the next sections, we describe in details the laser-based detection system depicted in Fig. 5.1. Each one of the frames in the figure is presented as follows: **coarse segmentation** is defined in Section 5.2.1.1; the β-skeleton procedure in charge of performing the **fine segmentation** is addressed in Section 5.2.1.2; the labeling procedure and **Procrustes analysis** are described in Section 5.2.2. After finding the labels and respective confidences, a **context analysis** is accomplished according to the procedure provided in Section 5.3.

5.2 From laser points to labels

After laser point segmentation, object detection is often done by extracting geometric features [Douillard 2007], [Mahlisch 2006], [Premebida 2007]. Conversely, we follow here a different direction based on a featureless approach. The goal is to cluster meaningful segments in a coarse-to-fine way, labeling them with a level of confidence given by a template matching procedure, based on Procrustes analysis [Joachims 1962].

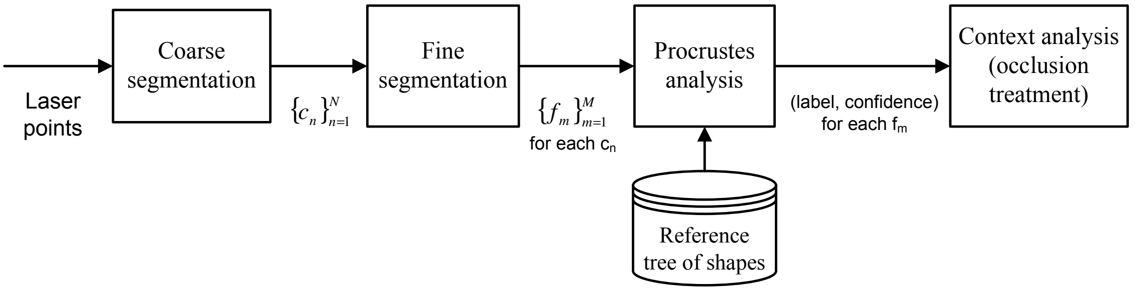


Figure 5.1: Outline of laser-based detection system. Each one of the process frames is detailed in the next sections: **coarse segmentation** is given in Section 5.2.1.1; the β-skeleton procedure in charge of performing the **fine segmentation** is addressed in Section 5.2.1.2; the labeling procedure based on **Procrustes analysis** is described in Section 5.2.2; after finding the labels and respective confidences, a **context analysis** is accomplished according to the procedure given in Section 5.3.

5.2.1 Segmenting points

5.2.1.1 Coarse segmentation

Let $\mathcal{L} = \{l_z\}$ be the number of points of a laser scan, where $z = 1, ..., Z$. After coarse segmentation, we have $\mathcal{C} = \{c_n\}$ coarse segments, where $n = 1, ..., N$. The set $\mathcal{C}$ is obtained by convolving the points in $\mathcal{L}$ with a kernel mask [-1,1], clustering those points at a distance of another point smaller than a threshold Δ. By assuming the size of a segment as the Euclidean distance between its endpoints, only segments $\in$ [0.25, 1] meters are considered as coarse segments. These values range from human body parts to large human silhouettes.

5.2.1.2 Fine segmentation

Let us note that inside a coarse segment c_n, there might be some "holes" (bigger spaces between groups of points) coming from the scanning process when the pedestrian is detected with body parts away from one another. This prevents any attempt to label c_n directly. To tackle this problem, we subsegment the set $\mathcal{C}$ into finer segments $\mathcal{F} = \{f_m\}$, where $m = 1, ..., 3$. The defined maximum of finer segments is 3 due to the adopted waist-level laser mounting (see Section 3.2).

The step of finding the set $\mathcal{F}$ is done by a β-skeleton method, a parameterized family of relative neighborhood graphs [Marchette 2004] proposed by Kirkpatrick and Radke [Kirkpatrick 1985]. This way, we have the following general definitions

Definition 1 *Let $V \subset \mathbb{R}^d$ be a vertex set. For any pair $(p,q) \in V \times V$, the set $\Lambda_{p,q}(\beta) = B((1-\beta/2)p+(\beta/2)q, (\beta/2)d(p,q)) \cap B((1-\beta/2)q+(\beta/2)p, (\beta/2)d(p,q))$ is called a **lune**, and $B(x, rad)$ is a circle of radius rad centered at x.*

Definition 2 *The β-skeleton, $G_\beta(V)$, with vertex set $V \subset \mathbb{R}^d$, is defined to be the graph on V with edge set E defined by lunes; thus $p, q \in E(G_\beta(V)) \Leftrightarrow \Lambda_{p,q}(\beta) \cap V = \varnothing$.*

It is noteworthy that:

- If $0 < \beta < 1$, $\Lambda_{p,q}(\beta)$ contains no points other than p and q.
- If $\beta = 1$, the circle of diameter (p,q) contains no other points than p and q.
- If $\beta > 1$, the union of two circles having diameter $\beta d(p,q)$ contains no points other than p and q.

In the last case, G_β is not guaranteed to be connected. It is straightforward to conclude that the value of β controls the sparseness of the graph. Hence, we conceived a clustering tendency index, T, to fit the best β to our goal. For this purpose, we expect that the vertices of $G_\beta \in \mathcal{F}$ have a degree equal to 1, and that the number of clusters is not greater than 3 (because of the adopted waist-level laser mounting). The clustering tendency index is defined as

$$T = log\left(\frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}(\gamma_i + \zeta_i)^{g_i+h_i}\right), \tag{5.1}$$

where $|\mathcal{C}|$ is the cardinality of the set $\mathcal{C}$, and the functions γ_i, ζ_i, g_i and h_i are given as follows:

$$\gamma_i = \frac{1}{K_i}\sum_{k=1}^{K_i}\sum_{j=1} e_{kj}\,, \tag{5.2}$$

where K_i is the number of clusters in a coarse segment i, and e_{kj} is the edge distance between two vertices with j indexing the number of edges inside a cluster k (fine segments);

$$\zeta_i = \begin{cases} \frac{1}{K_i-1}\sum\limits_{k=1}^{K_i-1} S_k, & \text{if } K_i > 1 \\ 0, & \text{otherwise,} \end{cases} \tag{5.3}$$

where S_k is the Euclidean distance between two fine segments;

```
input : $\{l_z\}_{z=1}^{Z}$ laser points $\subset \mathcal{C}$
output: $\{f_m\}_{m=1}^{M}$, $1 \leq M \leq 3$
for each $l_i$ from $l_1$ to $l_z$ do
    for each $l_j$ from $l_i$ to $l_z$ do
        if $l_i$ and $l_j$ satisfy Definitions 1 and 2 with $\beta = 2.1$ then
            addToAdjMatrix($l_i$,$l_j$)
        end
    end
end
```

Algorithm 4: Laser fine segmentation. $\beta = 2.1$ was the value chosen according the clustering tendency index.

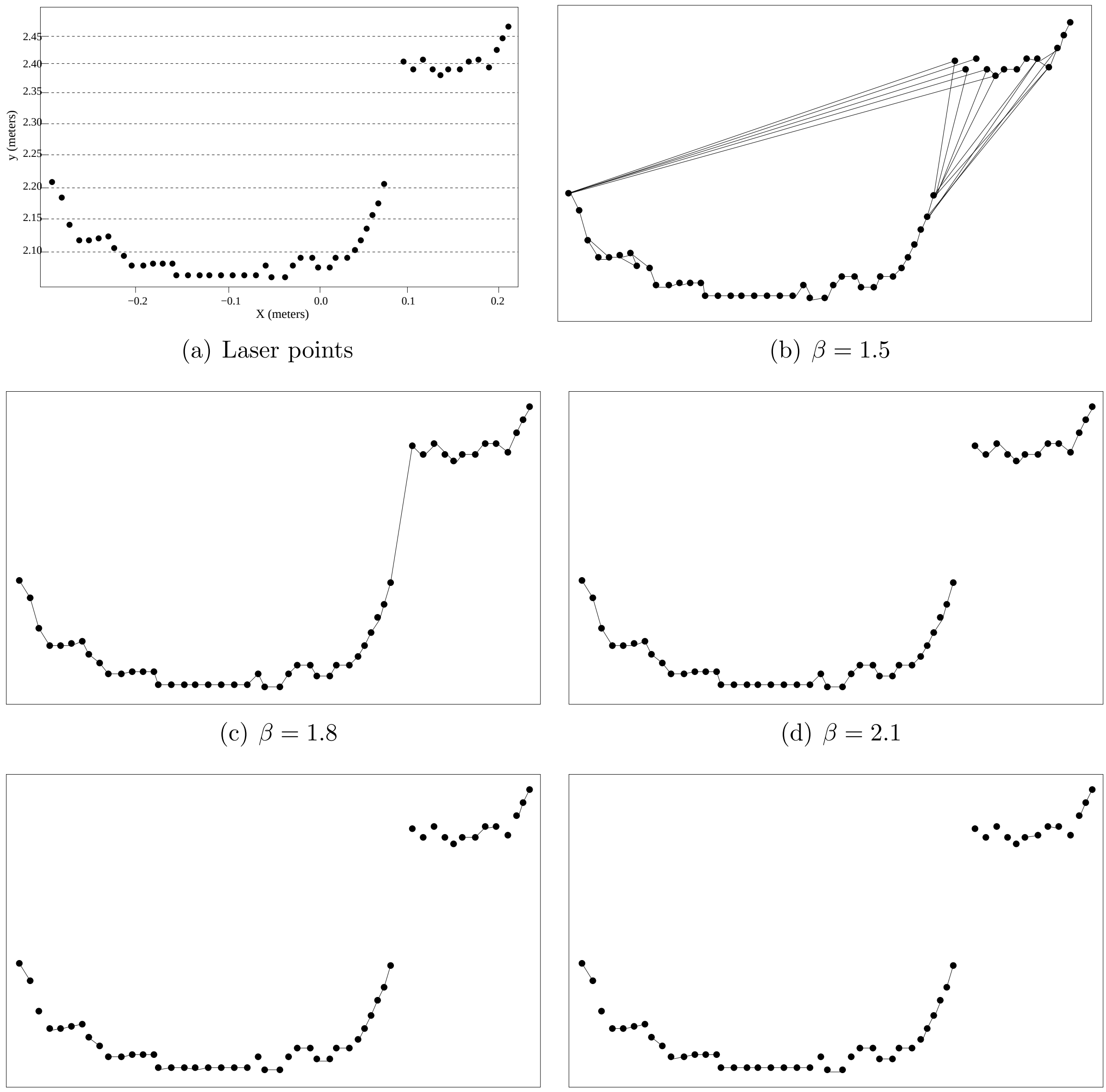
$$g_i = \begin{cases} 1, & \text{if } \max_{v_{ij} \in V}(deg(v_{ij})) = 0 \\ \max_{v_{ij} \in V}(deg(v_{ij})), & \text{otherwise,} \end{cases} \tag{5.4}$$

where $deg(v_{ij})$ is the number of edges incident to the vertex $v_{ij} \in V \subset \mathcal{C}$;

$$h_i = \begin{cases} 1, & \text{if } K_i \leq 3 \\ K_i, & \text{otherwise.} \end{cases} \tag{5.5}$$

The rationale behind function T is to penalize values of β which build random graphs whose degree of the vertices is other than 1, and the built graphs are too sparse. Hence, the goal is to take the lowest value of T. For a set of laser scans, β-skeleton graphs were computed with β varying in the interval [1,3] with a stride of 0.1, averaged over all laser scans. Figure 5.2 depicts a sample of laser points (Fig. 5.2(a)), and 5 resulting graphs (coarse segments) with some different β values. Figure 5.3 shows the results for the β values averaged over a training data set. $\beta = 2.1$ was the chosen value (see Fig. 5.2(d), for an example).

Algorithm 4 depicts the steps of the fine segmentation procedure in order to build a random graph for each coarse segment, saving it in an adjacency matrix. Although this procedure gives a clustering tendency, it can still fail in the clustering boundaries. However, rather than subsequently pruning the graphs at this level, this is done in the labeling stage.



(a) Laser points

(b) $\beta = 1.5$

(c) $\beta = 1.8$

(d) $\beta = 2.1$

(e) $\beta = 2.8$

(f) $\beta = 3.0$

Figure 5.2: Coarse-to-fine segmentation. In (a), a coarse segment corresponding to a human torso and hand. In (d), there are 2 fine segments, while, in (f), there are 23 fine segments. Note that, from (b) to (f), as β grows, the sparseness of the random graphs increases.

5.2.2 Labeling segments

In this stage, each fine segment $f_n \in \mathcal{F}$ is labeled as 4 different classes: "torso", "arm", "potential occlusion" and "noise". "Noise" segments are always discarded even if they fit into another class (by segment width), because it is composed of just two laser points.

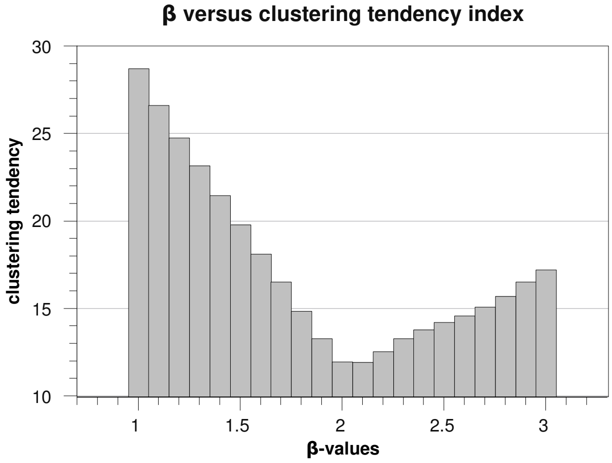


Figure 5.3: Cluster tendency: For each set of training images, β-skeleton graphs were computed with β varying in the interval [1,3], and stride equal to 0.1. The lower the value of T, the better. As a result, $\beta = 2.1$ was chosen.

Table 5.1: Relabeling rules

Seq. of labels in a coarse segment	**Relabeling to**
"potential" / "potential"	→ "torso"
"potential" / "torso"	→ "torso"
"arm" / "potential" / "potential"	→ "arm" / "torso"

Each label is initially given by measuring the width of the segments. It is subsequently relaxed with a confidence score, assigned by a shape matching procedure using Procrustes analysis. Only "torso" and "arm" segments have scores assigned. In this stage, it is possible to find wrong labels because of the boundaries on thresholding the coarse segments. To overcome that, a relabeling procedure was conceived based on the rules in Table 5.1. Relabeling relies on the view of an inconsistency on the semantic form of grouping the parts of a pedestrian silhouette. Note that, after all, final decisions will consider not only the label but also its confidence score.

To compute the confidence score of each label, a Procrustes analysis is performed, which matches the shape of a target segment to a hierarchy of reference shapes (Fig. 5.4), after filtering[1] translation, scale and rotation effects of each pair of laser points (reference and target). Procrustes is a statistical analysis of shapes, proposed by Huerley and Cattel [Joachims 1962], commonly used in biology field to match landmark data [Zelditch 2004].

[1] The concept of filtering translation, scale and rotation effects is equivalent to geometric transformations of the points into a common coordinate system, following the definitions of geometric morphometrics found in [Zelditch 2004].

Landmark data are corresponding points between two shapes, which are chosen manual or automatically. In our case, the landmarks are the laser points themselves. Next, the necessary definitions to perform Procrustes analysis [Dryden 1998] are presented.

Definition 3 *An $m \times m$ rotation matrix satisfies $\Gamma^T\Gamma = \Gamma\Gamma^T = I_m$ and $|\Gamma| = +1$. The set of all $m \times m$ rotation matrices is known as the special orthogonal group SO(m).*

Definition 4 *The Euclidean similarity transformations of a shape X are the set of translated, rotated and isotropically scaled X, such that*

$$\{\alpha X\Gamma + \theta^T : \alpha \in \mathbb{R}^+, \Gamma \in SO(m), \theta \in \mathbb{R}^m\}, \tag{5.6}$$

where α is the scale factor, Γ is the rotation matrix and θ is a translation m-vector.

To conform a target shape X_1 to a reference shape X_2, it is necessary to apply one transformation at a time. To filter out the translation effect from the laser points, centered landmarks, X_C, are obtained from the target landmarks X according to

$$X_C = CX\,, \tag{5.7}$$

where C is a translation to the centroid of X.

In case of scale filtering, it is necessary to standardize for size, which is done as follows

$$\Phi = \alpha X_C\,, \tag{5.8}$$

where Φ is invariant under translation and scaling of X, and is called the pre-shape of X, $\alpha = 1/\|X_C\|$, with $\|X_C\| = \sqrt{trace(X^T CX)}$.

Then, the full Procrustes distance, d_{proc}, where $0 \le d_{proc} \le 1$, between X_1 and X_2 with pre-shapes Φ_1 and Φ_2 is given by minimizing over rotations and scale to find the closest Euclidean distance between Φ_1 and Φ_2, such as

$$d_{proc}(X_1, X_2) = \left\{\ 1 - \left(\sum_{i=1}^{R} \lambda_i\right)^2\ \right\}^{1/2}\,, \tag{5.9}$$

where $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_{R-1} \ge |\lambda_R|$ are the square roots of the eigenvalues of $\Phi_1^T\Phi_2\Phi_2^T\Phi_1$.

The rotation that minimizes the distance between Φ_1 and Φ_2 is given by

$$\hat{\Gamma} = UV^T\,, \tag{5.10}$$

where $U, V \in$ SO(m) and $\Phi_2^T\Phi_1 = V\Lambda U^T$, with $\Lambda = diag(\{\lambda_i\})$.

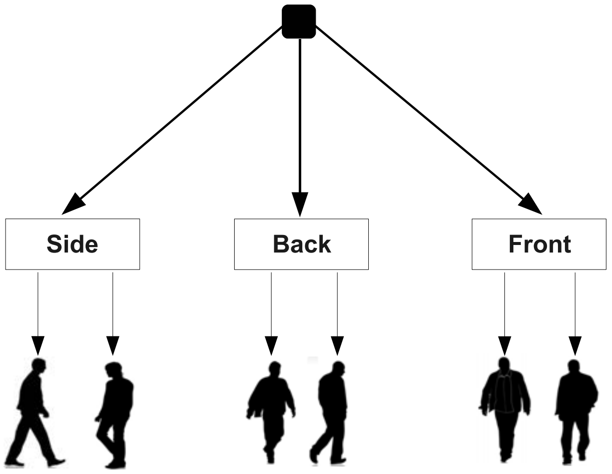


Figure 5.4: Tree of shapes: Pre-shape silhouettes of 10 different persons were averaged to be used as shape references. Back and front models were used with hands close and far from the side of the body, while side models were considered in left and right positions.

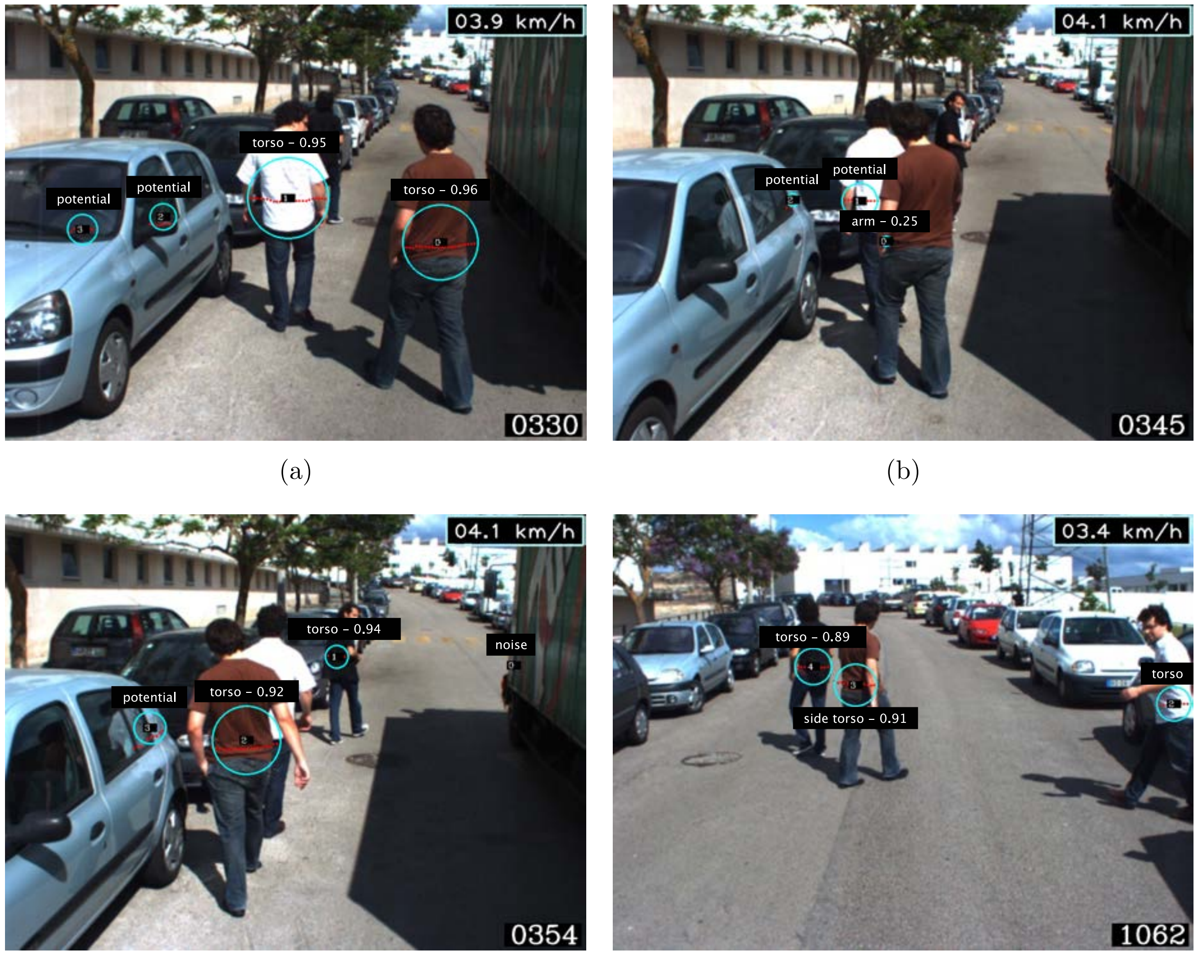
```
input  : $\{l_z\}_{z=1}^{Z}$ points from one scan and
         $\{xr_{ij}\}_{i=1,j=1}^{I,J}$ reference landmarks
output: A set of pairs $\{f_m, d_m\}_{m=1}^{M} \subset \mathcal{F}$, Dists
Dists ← initialize an array of Procrustes distances;
$\{c_n\}$ ← CoarseSegment($\{l_z\}$);
$\{f_m\}$ ← FineSegment($\{c_n\}$);
NeedRelabel ←Label($\{f_m\}$);
if NeedRelabel then Relabel($\{f_m\}$);
for each $f_m$ do
    for each $\{xr_i\} \subset \{xr_{ij}\}$ do
        if $|l_z| > |xr_i|$ then Extrapolate($l_s$);
        else if $|l_z| < |xr_i|$ then Shrink($l_s$);
        Dists ← Match($\{l_z\}$, $\{xr_i\}$)
    end
end
Dists ← max( Dists );
```

Algorithm 5: Laser segmenting and labeling

The scale that minimizes the distance between Φ_1 and Φ_2 is

$$\hat{\alpha} = \sum_{i=1}^{R} \lambda_i \,. \tag{5.11}$$

To match the target shapes with the reference shapes, a tree of laser segments was



(a)

(b)

(c)

(d)

Figure 5.5: Results of laser segmenting and labeling over a testing data set. On the right top and bottom of each image, information about the vehicle speed and frame number, respectively. In (a), two potential occlusions which will be semantically fused later, according to spatial relationships between objects. In (b), a very low confidence in hand labeling will take it to be discarded. In (c), an example of noise segment which is always discarded. In (d), an example of a side torso.

collected from persons with different silhouettes. The pre-shape of the collected segments were averaged over 10 samples. The tree of reference landmarks are comprised of sided, back and front shapes as depicted in Fig. 5.4. Back and front models were used with hands close and far from the side of the body, while side models were considered in left and right positions.

Procrustes analysis imposes equality of the two sets of landmarks. As equality is not guaranteed, that constraint is overcome by extrapolating or shrinking points in the target set. Since the shape of a pedestrian captured by the laser tends to hold a U-form, a parabolic regression suffices to obtain extra points, satisfactorily. Conversely, if the cardinality of the target set of landmarks is smaller than the cardinality of the reference set, the last points are discarded. The Procrustes distance for each segment is given by the maximum value over the distances found between the target landmarks and all the landmarks in the tree. All steps to segment and label the laser points are described in **Algorithm** 5, while some results of the algorithm over a testing data set are shown in Fig. 5.5.

5.3 Dealing with occlusion by spatial relationship between laser segments

In laser space, the occlusion problem is treated with a set of situation assertions defined from the geometry of the scene. These assertions rely inherently on partial segments, which are defined as

Definition 5 *A partial segment is a coarse segment which contains only one fine segment labeled as "arm" or "potential occlusion". This is so to contrast with a coarse segment comprised of fine parts (or of a "torso") which fully define a pedestrian, like, for example, "arm"/"torso" or "arm"/"torso"/"arm". A partial segment can initially be discarded, if it is not expected to be a pedestrian itself. However, in context, partial segments may represent strong hints to deal with occlusion.*

From the definition, if a partial segment is "near" from a full segment, it is a candidate for being an occlusion. The concept of "nearness" is used on a partial segment according to the angle between its end point and the start point of the closer full segment (Fig. 5.6). It is noteworthy that the partial segment must be after (in depth) its adjacent full segment to be an initial pedestrian candidate. The relationship of the attributes as well as their "nearness" is modeled as first-order logic in an MLN framework, which will be addressed in the next chapter.

Some results of occlusion detection are illustrated in Fig. 5.7. Figure 5.7(a) shows an example of potential occlusion ("potential(near)"), while 5.7(b) illustrates an example of arm ("arm(near)"). In both examples, the partial segments are occluded by full segments, being both considered in the fusion as hypothesized pedestrians. Figure 5.7(c) shows two cases where potential occlusions are correctly discarded ("potential discarded"), while Fig. 5.7(d) exemplifies a wrongly discarded pedestrian.

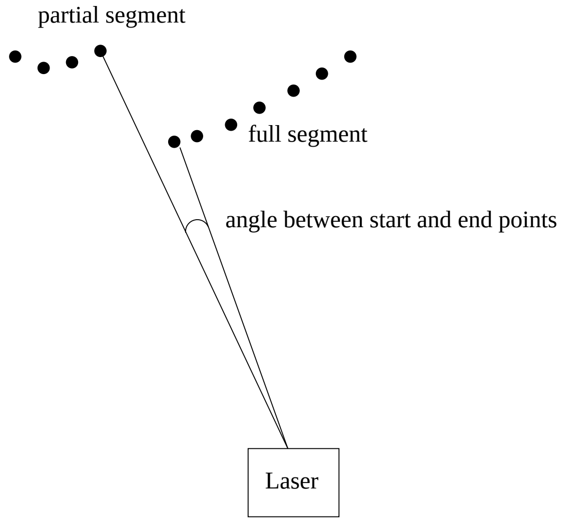
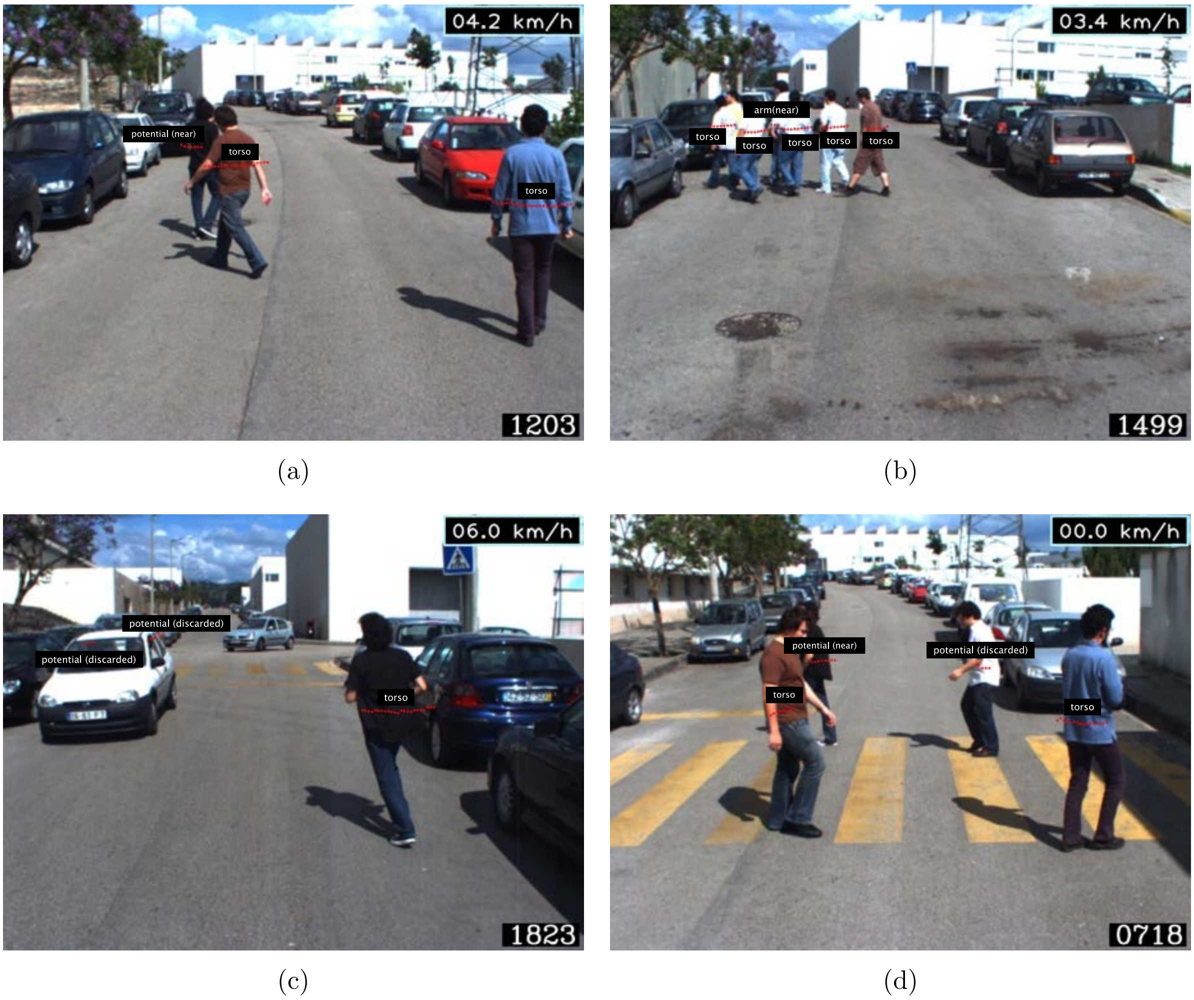


Figure 5.6: The "nearness" between two segments is defined as the angle (proportional to the laser angle resolution) given by the end and start points of the two segments, relative to the laser sensor. The "nearness" is only aplicable to partial segments in reference to adjacent full segments.

5.4 Conclusions

By treating partial segmentation, we are able to deal with the problem of object occlusion using laser. This approach is highly beneficial to keep many segments which could be easily discarded if only full segments were considered. The goal of this approach was to build a flexible pedestrian detector in laser space to be employed in highly dynamic environments, where objects are frequently crossing. This situation leads to many levels of occlusion, and by evaluating partial segments, it is possible to take this issue into consideration.

The proposed clustering approach is particularly interesting since there will be painless substitution of the algorithm if we decide to move to a multi-layer or 3D laser scanners. Although our laser-based object detector has been described in detail in this chapter, we postpone the experimental results and analysis to the next chapter, where we present not only our proposed semantic sensor fusion, but also how we integrate the sensor-driven detectors in the proposed framework (see Fig. 3.1).



(a)

(b)

(c)

(d)

Figure 5.7: Some examples of occlusions. In (a) and (b), it is shown examples of potential occlusion and arm, "near" a full segment. (c) shows two potential occlusions discarded because they do not satisfy the concept of "nearness". In (d), an example of a wrongly discarded "potential" pedestrian (second last pedestrian to the right); in this last situation, vision may help the detection, since our fusion system is not ROI-based.

Chapter 6

Semantic fusion of laser and vision

Contents

6.1 Sensor-driven detectors **71**

6.1.1 Image object detector 72

6.2 Foundations of Markov logic network **76**

6.2.1 First-order logic 76

6.2.2 Markov random field 77

6.2.3 Markov logic network 77

6.3 Semantically integrating the parts **80**

6.4 Experimental validation **81**

6.5 Conclusions **84**

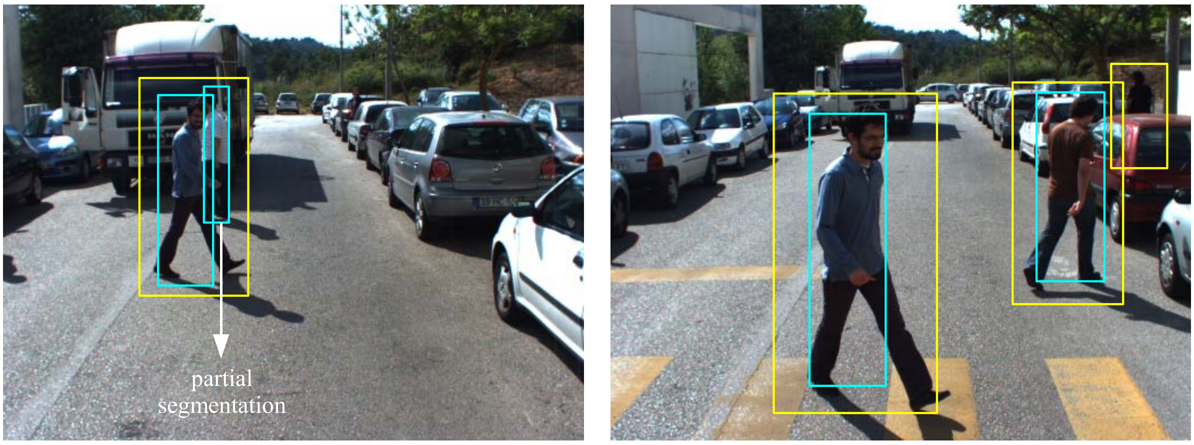
The symbolic view of things is a consequence of long absorption in images. Is sign language the real language of Paradise?

Hugo Ball

Existing methods of laser-vision fusion have been performed by ROI-based approaches. They are usually of two kinds:

1. Integration of sensor-driven classifiers (classifier-level fusion) or sensor-driven features (feature-level fusion), with IID assumptions.

2. After finding a hypothesized segment in laser space, the end points of this segment are then projected into the image, and an image classifier is applied to name the corresponding image region as object or non-object.



(a) (b)

Figure 6.1: Two scenarios where traditional laser and vision fusion systems would fail. The yellow bounding boxes are the result of an image classifier (laser independent), while the blue ones are the result of laser segmentation. In (a), an example of partial segmentation, which is not usually considered by a traditional laser segmentation system, is illustrated; hence, only the pedestrian closer to the camera could be detected by ROI-based approaches. In (b), pedestrians might be considered by the existing fusion methods, only where image classification and laser segmentation match.

Although the goal of those architectures seems to be clear-cut to speed up image object detection, they may fail in many situations. Figure 6.1 illustrates two scenarios where those approaches usually give wrong outputs. In Fig. 6.1(a), the partial segment would be discarded by traditional fusion methods, since it represents segments bellow a certain established threshold. In other words, laser segments, in traditional methods, are expected to be full, ranging over lower and upper thresholds, in order to be considered as a hypothesized object. Figure 6.1(b) shows a situation where only the vision system detected the occluded pedestrian. That pedestrian would be discarded by ROI-based fusion methods.

To overcome those issues, we propose here to integrate laser and vision in a semantic way. By semantic, we mean shifting from numeric values to symbolic representation by using situation assertions [Lambert 2006]. However, a question remains: *How can the symbolic representation acquire meaning?* We found an answer to that question through first-order logic (FOL). In a high level of abstraction, FOL are free-grammar deductive systems which can cope with very complex situations. The big limitation of FOL is, however, its main characteristic, that is, satisfiability. Inference in this type of deductive system must be hard constraint. In other words, if a set of FOL formulas does not satisfy a world (interpretation of assertions), then this world is not possible to exist.

To soften that constraint, an MLN was used [Richardson 2006]. This is a merge of Markov Random Fields (MRF) and FOL, where the nodes of the MRF are, indeed, the FOL formulas, while the links are the relationship between two nodes. Rather than relying on hard satisfiability, if a world does not satisfy a set of FOL formulas, then it is less likely to exist, but not impossible.

MLN is, indeed, the tip of the iceberg of our system, since we embody semantic information on many levels before using that final learning system. For that, we build a set of contextual assertions in order to cope with difficult object detection scenarios[1]. Our main goal is then to build a frame-by-frame classification system, based on the spatial relationship among the attributes of objects in the scene, constrained by contextual information. The main characteristics of the proposed fusion system are: treating partial segmentation, recovering depth information even when the laser segmentation fails to detect an object (allowing a vision classifier to detect an object, if the laser fails), and making the fusion not restricted to IID assumptions.

In this chapter, a description of our proposed semantic fusion is given. In addition, the foundations of the proposed detectors described in the previous two chapters are recalled in order to be properly integrated in the proposed framework. Although the aim is to detect pedestrians, the system can easily be adapted to any other types of objects. For that, it suffices a model of the constituent part of the target object.

6.1 Sensor-driven detectors

Within our fusion framework, the detectors applied to each sensor space are based on those described in Chapters 4 and 5. Our laser object detector entirely follows the description of the last chapter, while the structure of the image object detector was modified to attend the new requisites. Therefore, we address bellow the necessary changes in order to use HLSM-FINT in our system.

[1] Some of these assertions were discussed on Chapter 5 for laser segmentation.

6.1.1 Image object detector

6.1.1.1 Constraining object search

A typical image object detector uses a sliding window method to find a pedestrian in all image scales and positions[2]. Instead, we explore the 3D geometry given by the laser sensing space. There are twofold reasons for that:

1. Even if the laser classification fails, we are able to recover approximate depth information.

2. By constraining the detection to meaningful places, we are able to avoid many false alarms.

Let $\boldsymbol{\pi_1} = (0, -1, 0, dist_1)^T$ and $\boldsymbol{\pi_2} = (0, 1, 0, dist_2)^T \in \mathtt{R}^4$ be two planes in homogeneous coordinates given with respect to laser reference frame. These represent the ground (π_1), and a plane that crosses the camera optical center (π_2) (see our equipped vehicle with the on-board sensor in Fig. 3.2), which are parallel to the laser plane. This way, $dist_1$ (0.9m) represents the distance of the laser and the ground, and $dist_2$ is the distance between the center of the laser to the center of the camera lens ($dist_1 = dist_2$). A 3D sliding window procedure is defined with $\boldsymbol{\pi_1}$ and $\boldsymbol{\pi_2}$ as the top and bottom bounding planes, comprising a constrained viewport. Within that viewport, a window sized 1.0m × 1.8m is shifted onto horizontal and vertical directions in laser sensing space, ranging over 2m up to 20m, with a stride of 0.20m, in the space between π_1 and π_2 (see Fig. 6.2).

For those 3D sliding windows, it is assumed that the laser is always parallel to the ground. In the target application (see Chapter 3 to recall the details), this constraint will be broken whenever the car is moving toward a slope on the ground. In practice, slope change will not influence the system for a very long time. A way to relax this constraint will be investigated in the future.

Note that after projection of the 3D sliding windows by using the Zhang and Pless' calibration procedure [Zhang 2004], only the 2D windows lying on the image frame are used. For each window, the depth information is kept for posterior reasoning (estimated depth information when the laser fails). Figure 6.3 shows an example of the windows after image projection.

[2] Efficient subwindow search (ESS) is a recently proposed searching method with sub-optimal results [Lampert 2007]. In the future, we intend to use this method as an alternative to sliding window method, since it is faster, and also more robust.

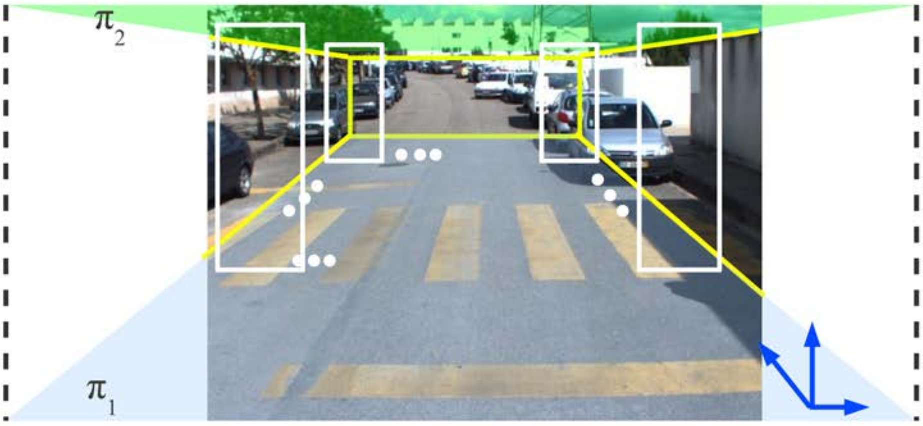


Figure 6.2: A window sized 1.0m × 1.8m is shifted onto horizontal and vertical directions in laser sensing space, with a stride of 0.20m, ranging over 2m up to 20m in depth. The searching area is constrained by the viewport formed by the planes π_1 and π_2 (parallelepiped).

6.1.1.2 Parts-based HLSM-FINT

Previously, HLSM-FINT was used monolithically. To improve the detection rate in face of occlusion situations, we used the INRIA data sets (see Table 2.1) to train a parts-based detector over two parts of the human body (Fig. 6.4), which are prone to be occluded in real urban scenarios. When pedestrians are far, this choice also avoids parts being lost as it can happen in limbs-based approaches. The 54 × 108 pixel reference window has an upper body with a height of 34 pixels, starting at position (0,0), and a lower body with a height of 24 pixels, starting at location (0,48), in reference to the origin of the detection window.

After applying the parts-based HLSM-FINT to each of the projected sliding windows, an object can be bounded by many windows. Hence, a non-maxima suppression algorithm is used to prune extra windows[3]. Here, depth information of each window is kept in case the laser detection fails.

Algorithm 6 describes the non-maxima suppression algorithm. The overlapping coefficient (line 3) is given by Eq. (4.16). The parameter `MinCluster` (line 5) defines the number of remaining windows after suppression.

Some image samples classified by our detector are illustrated in Fig. 6.5. Crowded

[3] This algorithm shares the same principle of that one used in Section 4.4.7 to evaluate the ensemble detectors over a sequence of frames.

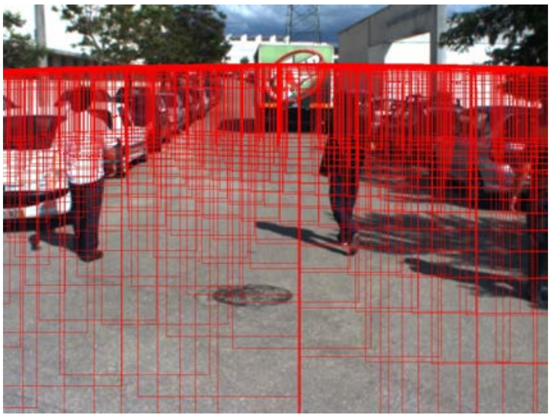


Figure 6.3: A window is slid in laser sensing space onto horizontal and vertical directions. Only those projected planes lying on the image are used. Depth information of each 3D window is kept for posterior reasoning.

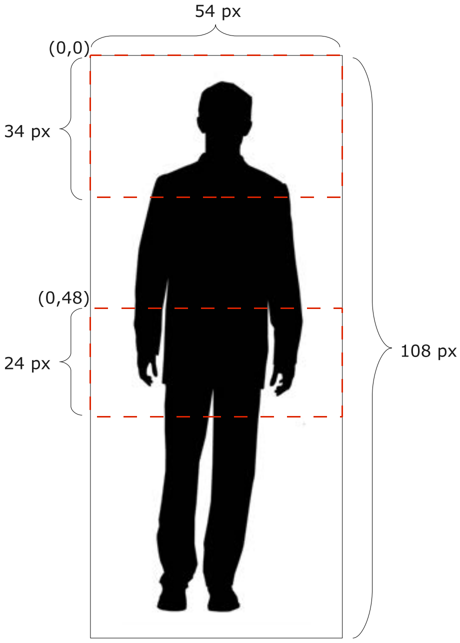
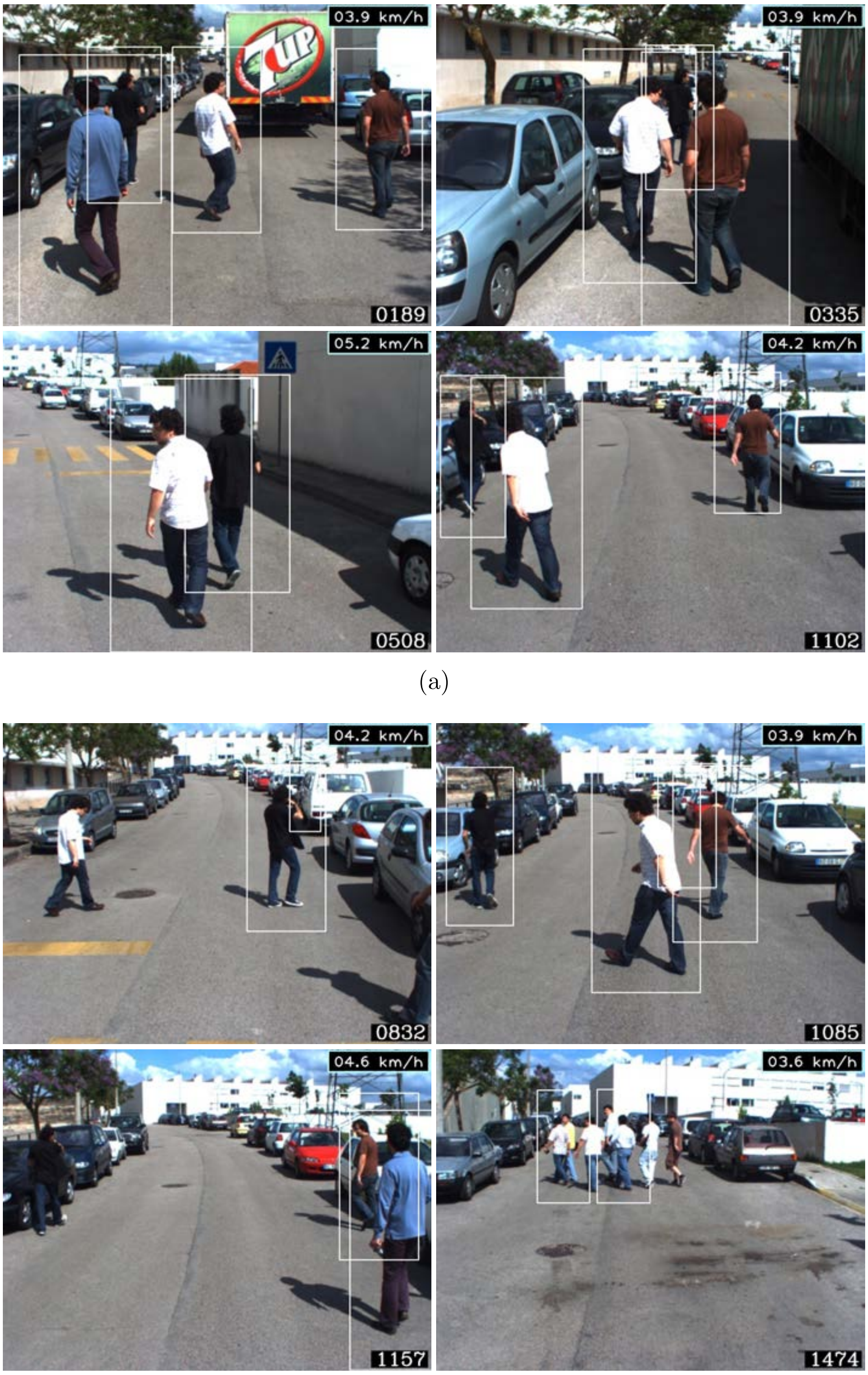


Figure 6.4: Pedestrian window (54x108 pixels) broken into two main parts: upper body (34 pixels) and lower body (24 pixels), covering shoulder and head, and the waist zone, respectively.

scenes are difficult for our image classification system as it is shown in the right-most image in Fig. 6.5(b), due to the inherent resolution problem of the sliding window procedure. Crowded scenes will be addressed more coherently by means of contextual information with the introduction of semantic fusion, which is described in the next sections.



(a)

(b)

Figure 6.5: Samples classified by the parts-based HLSM-FINT in the 3D sliding window framework. In (a), some difficult objects are successfully detected by our parts-based ensemble detector, while in (b), it is illustrated some detection failures.

```
input  : WindowArray $\leftarrow$ A set of detection windows
         ConfidenceScores $\leftarrow$ A classifier score for each detected window
output : PrunnedWinArray $\leftarrow$ Pruned windows

sortWindows(ConfidenceScores);
for each sorted window do
    OverlapCoef $\leftarrow$ Computing overlapping coefficient of each window against all
    if OverlapCoef $\geq$ MinThreshold then PrunnedWinArray $\leftarrow$ saveWindow();
end
if The number of windows in PrunnedWinArray $\geq$ MinCluster then
    Discard windows with smaller confidence scores
end
```

Algorithm 6: Non-maxima suppression algorithm in charge of pruning extra bounding boxes around the detected object.

6.2 Foundations of Markov logic network

The comprehension of MLN involves, in the beginning, understanding some of the main concepts of FOL and MRF. After that, the principles of MLN come straightforwardly by confining both paradigms into a representative framework. Hence, the main topics which structure MLN are addressed separately, followed by a final definition of the method and its application within our framework.

6.2.1 First-order logic

A first-order knowledge base (KB) is composed of a set of formulas in FOL. In turn, formulas can be formed by three types of symbols: constants, variables, and predicates. Constants represent an object in the universe of interest (e.g., `Pedestrian`). Variables range over objects. Predicates represent relationships among objects, or assign attributes to objects (e.g., `Part`, `Near`). Constants, variables and predicates are *terms* of the formulas. An *interpretation* assigns a symbol to each object in the universe of interest. It is noteworthy that a predicate can be a query, if it is desirable to make inferences from that (arguing something) given evidence predicates, where the knowledge is assigned.

FOL also makes use of quantifiers (universal quantifier ($\forall$) and existential quantifier ($\exists$)) to define the domain of a variable. An *atomic formula* or *atom* is a predicate applied to a tuple of terms. A *ground atom* is a predicate which contains constants assigned by possible variables. In FOL, if a world violates one formula, then this world is impossible to occur.

It is important to notice that it is usually difficult to model a complex problem just by using formulas which are always true or false. A set of FOL formulas somewhat defines only a fraction of the relevant knowledge. This way, inference on pure FOL may suffer of limited applicability when the world is not hard defined.

6.2.2 Markov random field

MRF is an undirected graphical model comprised of a set of nodes, with each node corresponding to a variable or a group of variables [Bishop 2006]. An underlying concept of MRF is graph clique, which is defined as a subset of nodes in the graph such that there is a link between all pairs of nodes in the subset. That concept actually brings the idea of locality, which is the core for inference methods in MRFs. In practice, an MRF is a joint distribution of a set of variables $X = \{x_n\}_{n=1}^{N}$, written as

$$P(X = x) = \frac{1}{Z} \prod_{C} \phi_C(x_C)\,, \tag{6.1}$$

where $\phi_C(x_c)$ is a potential function over the cliques of the graph, and Z is a normalization constant, given by

$$Z = \sum_{x} \prod \phi_C(x_C)\,. \tag{6.2}$$

The choice of potential functions is not restricted to those that have a specific probabilistic interpretation as marginal or conditional distributions, since the partition constant Z can be used to normalize $P(X = x)$, appropriately.

6.2.3 Markov logic network

The first idea of MLN [Richardson 2006] is to soften the FOL constraints, that is, when a world violates a formula, it becomes less probable, but not impossible. To this end, each formula in a KB has a weight which represents how strong a formula is for a world. The higher the weight is, the bigger the probability for a world to be satisfied. The main goal of MLN is thus to unify the fundamental advantages of FOL and MRF, dealing at the same time with complexity and uncertainty. This way, let us present the main definitions of MLN.

Definition 6 *A first-order MLN is a set of pairs $(\mathcal{Q}_i, w_i)$, where $\mathcal{Q}_i$ is a formula in FOL and w_i is a real-number weight. Each $\mathcal{Q}_i$ is a node of a MRF.*

Given different constants, a first-order MLN will produce different MRFs, which are called ground MLNs. Each ground MLN varies in size but it keeps regularities in structure and parameters, given by the first-order MLN. Rather than defining the ground MLN as the form (6.1), and because we are restricted to potential functions which are strictly positive, it is more convenient to write it as a Boltzmann distribution (exponential representation), such that

$$P(X = x) = \frac{1}{Z} exp\left(\sum_i w_i \eta_i(x_i) \right) , \tag{6.3}$$

where Z is now equal to $\sum_{x \in X} exp(\sum_i w_i \eta_i(x_i))$, $\eta_i(x)$ is the number of true groundings of $\mathcal{Q}_i$ in x_i, which is, in turn, the state of the ith atom in $\mathcal{Q}_i$.

The weights of an MLN can be learnt or hand-crafted. Weight learning can be performed generative or discriminatively. In our framework, we used a discriminative approach, based on a voted perceptron weighted satisfiability solver, which is demonstrated to outperform generative approaches [Parag 2005]. Discriminative learning is performed by using ascent method over the gradients of the conditional log-likelihood, given by

$$\frac{\partial}{\partial w_i} \log P_w(\psi|\varepsilon) = n_i(\psi, \varepsilon) - E_w[n_i(\psi, \varepsilon)] , \tag{6.4}$$

where ψ is a query predicate, ε is an evidence predicate, $E_w[n_i(\psi, \varepsilon)]$ is the expectation over the number of true groundings of formula i according to the MLN, approximated by a maximum a posteriori (MAP) inference. A training data set was used to learn the weights, w_i, of our MLN (#Seq. 1 in Table 3.1), by means of the Alchemy library [Richardson 2006][4].

For each step, t, in the gradient ascent method, $w_{i,t}$ is found by

$$w_{i,t} = w_{i,t-1} + \eta \frac{\partial}{\partial w_i} \log P_w(\psi|\varepsilon)|_{w_{t-1}} , \tag{6.5}$$

where η is a learning rate.

After training an MLN, inference is performed by a combination of Markov chain Monte Carlo (MCMC) and SampleSAT algorithms, called MC-SAT [Poon 2006]. As an MRF is ultimately represented by a joint probability (see Eq. 6.1) of potential functions, the inference in graphical model has the aim of finding marginal probabilities. This marginalization process is proceeded by computing the conditional probabilities of a node given its Markov Blanket. To sample the variables of a node, an MCMC algorithm is

[4] Available in http://alchemy.cs.washington.edu/

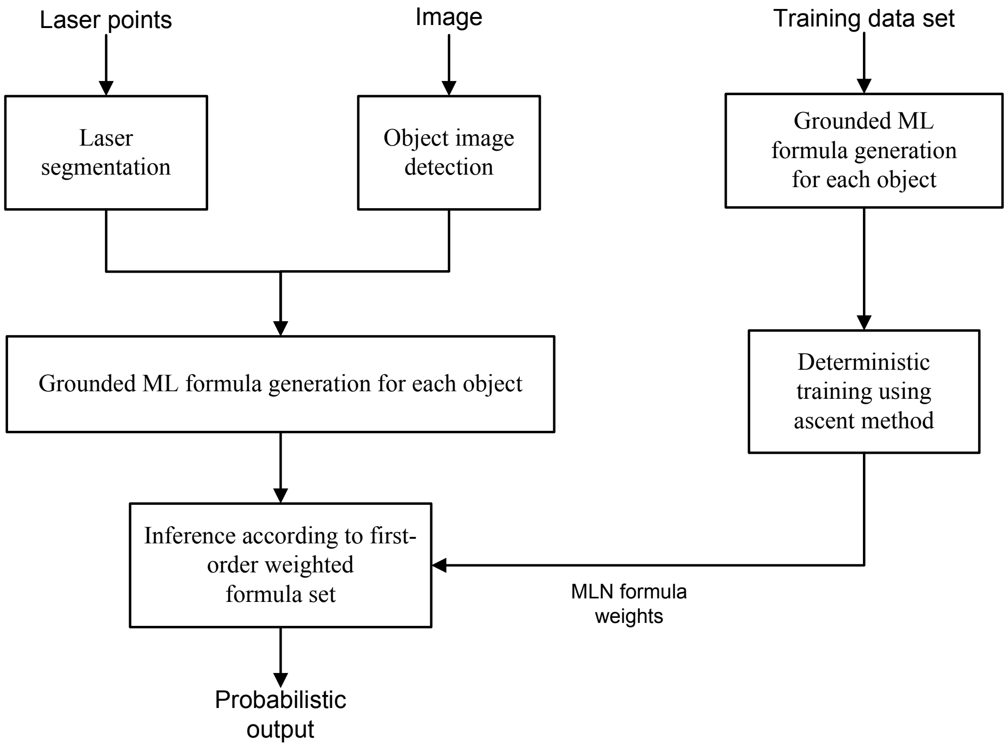

Laser points
Image
Training data set
Laser segmentation
Object image detection
Grounded ML formula generation for each object
Grounded ML formula generation for each object
Deterministic training using ascent method
Inference according to first-order weighted formula set
MLN formula weights
Probabilistic output

Figure 6.6: MLN framework. Either in the training or in the inference stages, a set of grounded ML formulas must be generated according to the FOL formulas in Table 6.1. After training, weights are given to each one of the FOL formulas in order to perform inference over the grounded formulas generated by the rules of laser segmentation and image classification. At the end, a probabilistic output is provided for each object found in the scene.

usually utilized through Gibbs sampling. For large domains, however, MCMC applied to MLNs is intractable, since this latter one needs probabilistic and logic inference, which are #-complete and NP-complete, respectively. To overcome this problem, Poon and Domingos [Poon 2006] proposed a new approach based on MC-SAT, which applies slice sampling to MLN, using SampleSAT to sample a new state given its auxiliary variable. SampleSAT algorithm samples solutions uniformly, at each iteration, applying a WalkSAT step with probability p and a simmulated annealing step with probability *1-p*.

Table 6.1: First-order formulas used in the semantic fusion. An object "*o*" can be either an image window or a laser segment.

No.	First-order logic	Description
1	∀o, Person(o)	Query over object *o*
2	∀o∀t∀p, Part(id,t,p)	Object *o* with part of type *t*, in position *p*
3	∀o,∀t, Unique_Part(o,t)	Object *o* of type *t*
4	∀o, Near(o)	Object *o* near by other object
5	∀o, HighScore(o)	Object *o* with small Procrustes distance
6	∀o, View(o,b)	Object *o* with a body part *b*
7	∀o, Unique_Part(o,‘‘potential’’) ∧ Near(o) ⇒ Person(o)	If an object *o* has any potential occlusion segment and it is near by any other object, it is a pedestrian
8	∀o, Unique_Part(o,‘‘arm’’) ∧ Near(o) ⇒ Person(o)	If an object *o* has only one arm segment and it is near by any other object, it is a pedestrian
9	∀o, View(o,‘‘upper’’) ∨ View(o,‘‘lower’’) ⇒ Person(o)	If an upper or a lower part of an object *o* is viewed, it is a pedestrian
10	∀o, Part(o,‘‘arm’’,i) ∧ Part(o,‘‘torso’’) ∧ HighScore(o) ⇒ Person(o)	If an object *o* has arm and torso with Proc. distance smaller than 0.25, it is a pedestrian
11	∀o, Part(o,‘‘arm’’,i) ∧ Part(o,‘‘arm’’) ∧ HighScore(o) ⇒ Person(o)	If an object *o* has arm and arm with Proc. distance smaller than 0.10, it is a pedestrian
12	∀o, Unique_Part(o,‘‘torso’’) ∧ HighScore(o) ⇒ Person(o)	If an object *o* has only a torso and a Proc. distance smaller than 0.10, it is a pedestrian
13	∀o∀i∀j, Part(o,‘‘arm’’,i) ∧ Part(o, ‘‘potential’’,j) ∧ HighScore(o) ⇒ Person(o)	If an object *o* has arm and potential segments and it has Procrustes distance smaller than 0.20, then it is a pedestrian
14	∀o∀i∀j∀l,Part(o,‘‘arm’’,i) ∧ Part(o,‘‘potential’’,j) ∧ Part(o,‘‘arm’’,l) ∧ HighScore(o) ⇒ Person(o)	If an object *o* has two arms and one potential, and it has a Procrustes distance smaller than 0.20, it is a pedestrian
15	∀o∀i∀j∀l,Part(o,‘‘arm’’,i) ∧ Part(c,‘‘torso’’,j) ∧ Part(o,‘‘arm’’,l) ⇒ Person(o)	If an object *o* has 2 arms and one torso, then it is a pedestrian

6.3 Semantically integrating the parts

In laser space, the occlusion problem is tackled by a set of situation assertions defined from the geometry of the scene, as described in the last chapter. These assertions rely inherently on partial segmentation. As a result, if a partial segment is "near" to an adjacent full segment, it is a candidate for being an occlusion (refer to Section 5.3 for more details). In image space, a parts-based classifier is applied to the windows of a sliding window procedure performed in the laser sensing space, and projected to the image. To integrate all object attributes (parts) and classifiers in a semantic way, we manually built a first-order KB using MLN language, learning its weight discriminatively.

Figure 6.6 illustrates the relationship of the components of the MLN framework. Grounded ML formulas are generated according to the rules defined by the FOL formulas presented in Table 6.1. After finding the weights for each FOL formula in the table, the inference is accomplished according to these learnt weights. Finally, a probability of being a pedestrian is given for each object in the scene.

The FOL formulas are summarized in Table 6.1. An object "*o*" can be either an image object or a laser object. An object is represented by a unique sequential number (generated in grounded ML formulas) as objects are detected. As mentioned before, these formulas will structure a grounded MLN.

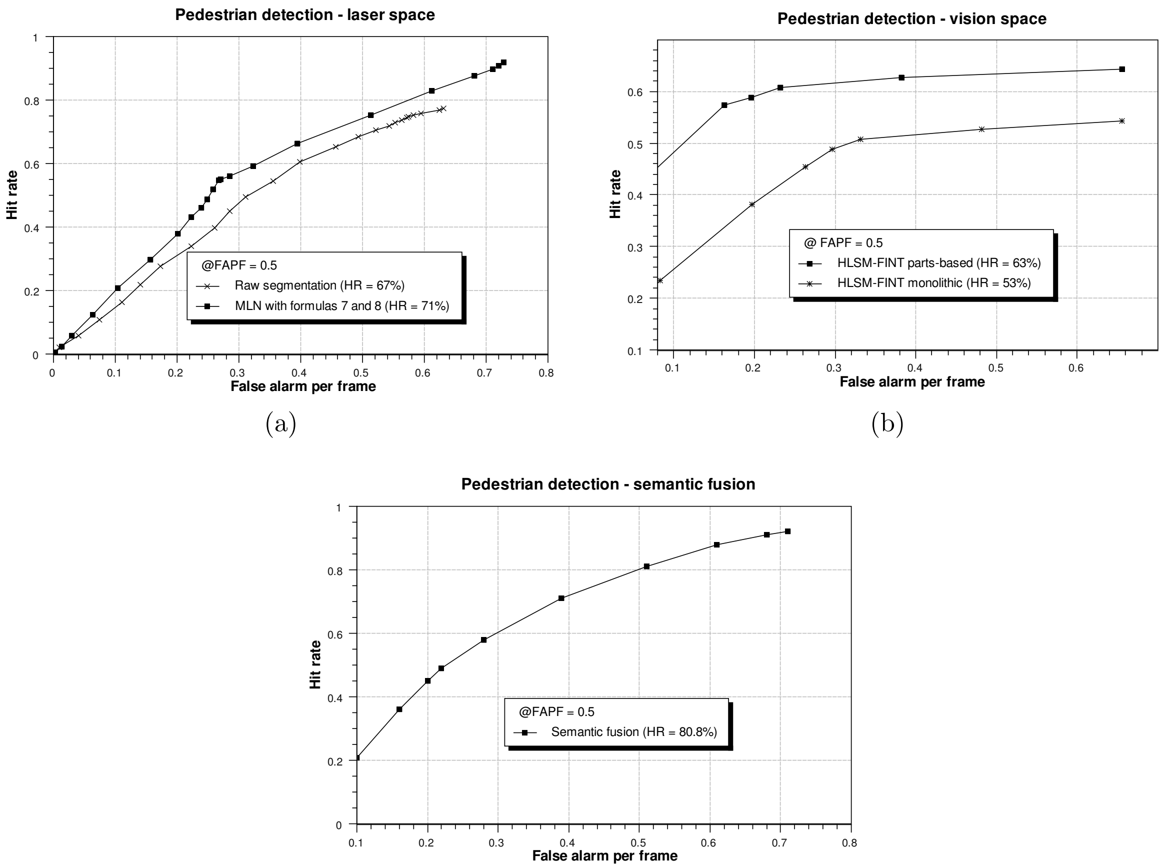
Formula 9 describes the problem of occlusion in the image space. Actually, by adding formula 9 as a disjunction in the formulas 10 to 15, it also works as the core of the laser-vision fusion. In other words, if for a given object o, this object is only seen in the image (upper or lower body), formula 9 will be considered alone, otherwise if a laser window overlaps an image window, it will be responsible for increasing the weight of formulas 10 to 15. If we find a torso as a unique part (formula 10), that is, there was just one part f_m, then a high confidence score (low Procrustes distance) is necessary to validate it as a pedestrian. Similarly, when a coarse segment, c_n, is formed by a potential occlusion (formulas 11 and 12), it means that the segment could have not been labeled appropriately due to the limit boundaries in the coarse step. In this case, the score of the coarse segment provides a hint to establish it as a pedestrian or not. The last formula represents the highest evidence to assign a coarse segment to a pedestrian, because the segmentation was hopefully perfect, not being needed to analyze the score of the segment.

6.4 Experimental validation

The collected data sets (see Table 3.1) were used to train and test the proposed method. Sequence #1 was used to train the MLN, and to select the clustering tendency index (Section 5.2.1.2), while Seq. #2 was used in the validation of the proposed semantic fusion. The detection performance was assessed by receiver operating characteristics (ROC) curves. To train the MLN, we used the Alchemy library.

ROC curves were built for the classification methods in each sensor space and for the proposed semantic fusion (see Fig. 6.7). The operating points of all curves were chosen to be at 0.5 false alarm per frame (FAPF). As concerns laser, the points were first segmented ($\Delta = 0.25$, for coarse segmentation), being noise segments discarded. Torso and arm were thresholded according to their Procrustes distances, used to score the fine segments (refer to the description in Table 6.1 of the formulas 10 to 14). After that, the first-order formulas 7 and 8 were applied. The results are shown in Fig. 6.7(a).

In laser detection, two experiments were accomplished in order to evaluate the detection performance: raw segmentation (done by using only the segmentation procedure described in Chapter 5), and the use of MLN formulas 7 and 8, presented in Table 5, over the raw segmentation. The former achieved an HR of 67%, while the later had 71% of HR. In image classification, Fig. 6.7(b) shows the results for monolithic (53% of HR) and



(a)

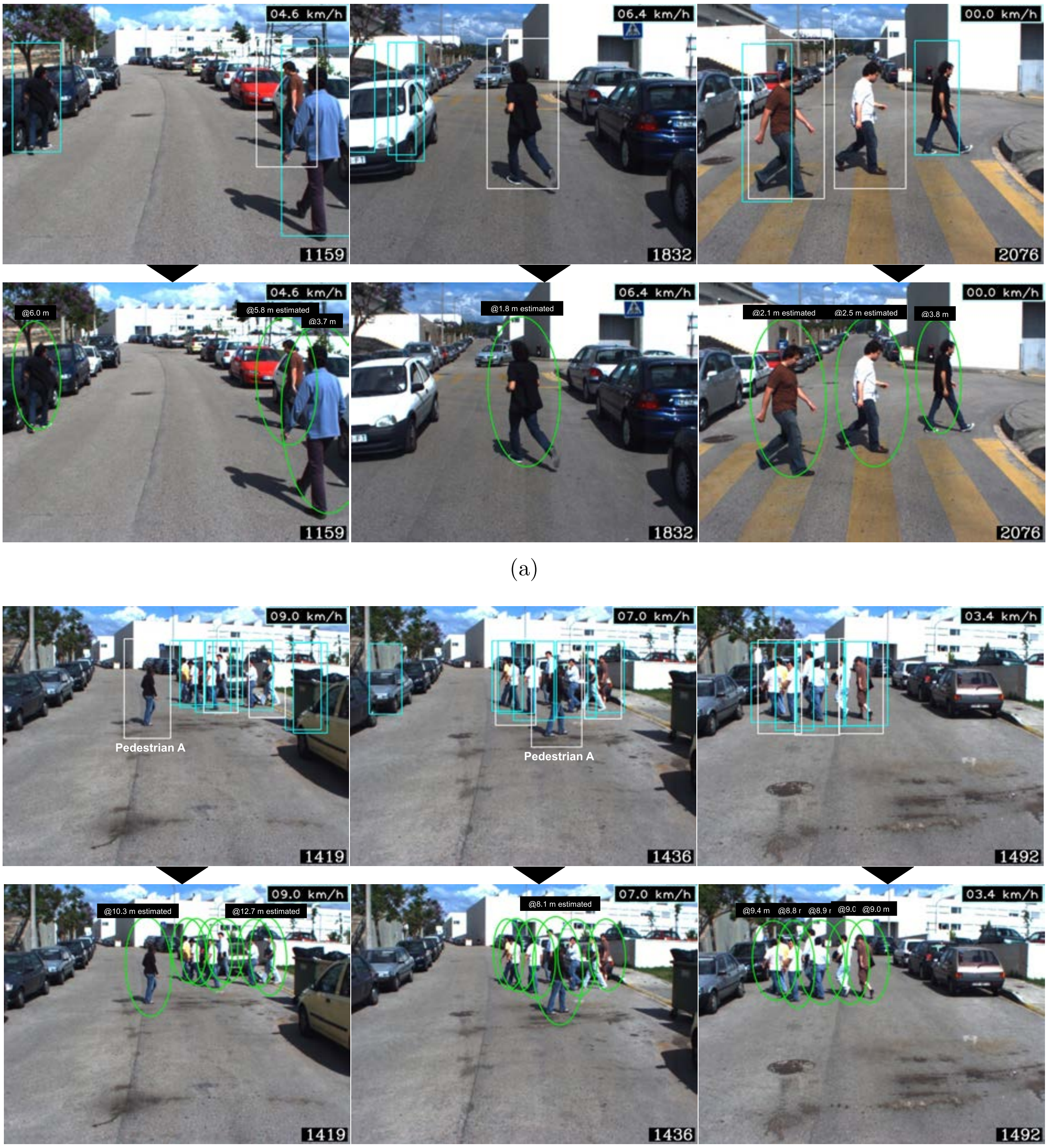
(b)

(c)

Figure 6.7: ROC curves of pedestrian detection regarding each sensor space and semantic fusion. The operating points in the curves were chosen to lie on 0.5 FAPF for all curves. (a) Results on laser raw segmentation and using first-order formulas 7 and 8. (b) Image classified with monolithic and parts-based versions of HLSM-FINT. In (c), results of the semantic fusion composed by our parts-based classifiers and all first-order formulas in Table 6.1.

parts-based HLSM-FINT (63% of HR). Although our detector reached high performance in [Oliveira 2010b], tested over DaimlerChrysler image data sets, here it was challenged with a more difficult data set, with a considerable number of occlusions and real-life situations. The results show that parts-based HLSM-FINT has a better performance, in this new data set.

Finally, Fig. 6.7(c) shows that the proposed method achieved 80.8% of HR, increasing the performance of our parts-based detector by 17.8 percentage points, and the performance of the MLN-based segmentation by 13.8 percentage points. Table 6.2 shows the summary of all results.



(a)

(b)

Figure 6.8: Samples of resulting semantic fusion. In (a) and (b), first rows are the result of the individual classification in each sensor space (the white bounding box is given by an image detector, while the blue one is given by a laser segmentation), and the second rows are the fusion result. In (a), a set of results where the laser fails and the image detector complements the recognition process. Note that the depth information when laser fails is estimated from the 3D searching windows. In (b), a highly dense occlusion scene.

Table 6.2: Summary of the final results

Classification method	**Hit rate**
HLSM-FINT (monolithic)	53.0%
HLSM-FINT (parts-based)	63.0%
Laser segmentation	67.0%
Laser segmentation with formulas 7 and 8 (Table 6.1)	71.0%
Semantic fusion	**80.8%**

Figure 6.8 illustrates some results of the semantic fusion. The first rows in Figs. 6.8(a) and 6.8(b) correspond to detection results in each sensor space, while the second rows show the resulting fusion. It is noteworthy that when only the output of the image detector is chosen by the fusion method, then an estimated depth information of the pedestrians is given. Some results of synergistic detection in laser and image spaces are shown in Fig. 6.8(a). The left-most image shows an occlusion situation wherein the parts-based detector succeeds, increasing the detection rate of the fusion method. Figure 6.8(b) shows a set of highly dense occlusion images, showing the effectiveness of our segmentation approach.

It was also observed in the two left-most image of the first row in Fig. 6.8(b) that the laser segmentation failed most of the times on “Pedestrian A”, who was crossing the street in front of the group of persons, providing only few points due to the low reflectivity of the black shirt in a far distance[5]. This problem is inherent to the laser scanner used, and it can be tackled with a more accurate sensor. In this case, our image detector was effective to improve the accuracy of the recognition.

6.5 Conclusions

Our proposed fusion system is not constrained by IID assumptions, and overcomes limitations found in existing fusion methods, such as: treating partial segmentation and recovering depth information when the laser segmentation fails. For that purpose, a parts-based detector along with some contextual assertions were essential to draw our fusion framework. The image classification approach was done by projections of 3D sliding windows applied to the laser sensing space. This was particularly important in recovering the depth information when the laser fails. Figure 6.9 illustrates the resulting fusion with our proposed method over the two scenarios depicted in Figs. 6.1.

An MLN framework provided a comprehensive framework to perform the final fusion,

[5] For SICK laser used in the experiments, above 10 meters, reflectivity goes to 10% in matt black materials [SICK 2006].

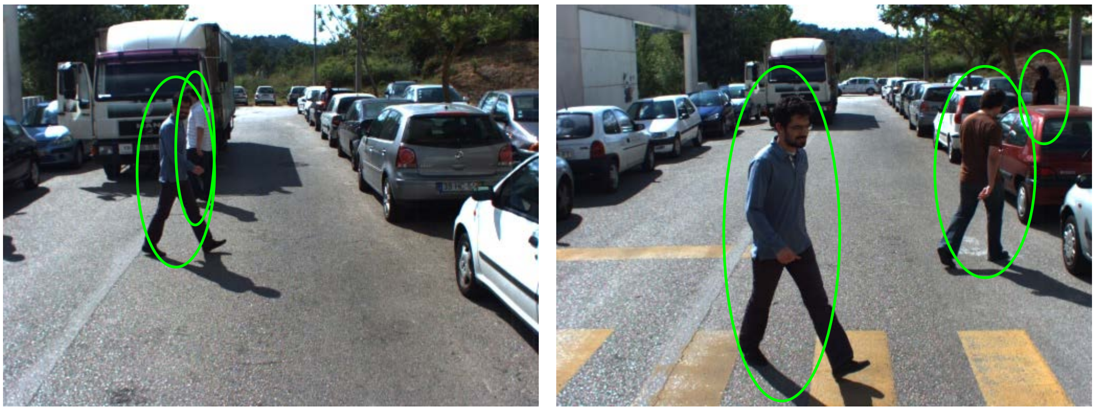


Figure 6.9: The results of our proposed fusion method applied to those scenes of Figs. 6.1.

addressing the fusion problem through a symbolic representation and the spatial relationship between the object attributes. Our proposed system increased the worst individual detection by 17.8 percentage points, reaching 80.8% of HR, with 0.5 FAPF, in a difficult data set. These results demonstrate how effective the proposed method is on sensing difficult situations, mainly with occluded objects.

Part III

Discussion and conclusions

Chapter 7

Discussion

Contents

7.1 Effective detection systems in ITS 89

7.2 Toward system implementation 91

7.3 Future directions 91

The oldest, shortest words – "yes" and "no" – are those which require the most thought.
Pythagoras

Our proposed work represents an effort to go one step further into the problem of object detection based on sensor fusion. Nonetheless, there is still a lot of work to do to turn it into an effective system applied to ITS. In this chapter, we remind our main contributions, analyzing them under the perspective of online implementation and performance in real-life applications. Finally, we draw our goals to future research in the fields of pattern analysis and ITS.

7.1 Effective detection systems in ITS

Let us consider a perfect detection machine applied to autonomous or driver assistance systems. Such a machine should ideally have the following characteristics:

- It must detect all objects in the FOV of the sensors.
- Its processing time must be short enough to detect objects considering not only the ego-vehicle[1] speed, but also the target-object velocity. In order words, in a real

[1] Where the detection system is onboard.

application, the information about an eminent collision must be sent immediately to the controller (driver or machine).

- It must provide accurate information about spatial localization of the target objects.

As it can be noticed, that list can not be accomplished by a so-called intelligent machine, nowadays. We can soften the level of that list demand, pointing out more realistic features to make object detection systems to be at least suitable for some real applications. And these are: To feature on-the-fly time processing according to the application, and to have the number of false alarms approaching to zero. For "on-the-fly processing time", we mean detecting the object of a scene in reasonable time, giving time for the driver or the autonomous machine to be warned, stopping the vehicle before colliding.

With respect to the performance, the higher the number of false alarms, the worser will be the vehicle's navigation, since the vehicle would be driven unsafely with a lot of stop-and-go maneuvers. On the other hand, even if the number of false alarms is low, having an equally low hit rate is not useful, making the system incapable to aid the driver or to ride the vehicle accordingly. In this regard, the main goal of our proposed detection system was to keep the number of false alarms still feasible to practical applications, not dropping the hit rate. However, our final result has indicated that a false alarm per frame of 0.5 is still high for a real-life application, which means that the system will present a false alarm in each two frames. This high value of false alarms can also be found in state-of-art systems [Leibe 2005], [Leibe 2007], [Andriluka 2008], designed for real-life application. Although those detection systems are evaluated on crowded scenes, the number of frames and annotated objects are much lower than in our work, despite the complexity of those architectures are much higher, and the applicability is much more broader. This issue indicates that this topic is still open for contributions, however, it is worth reminding that there is a trade-off between detection rate and computational cost in order to make a system suitable for real-life applications.

Another aspect of effective implementation of object detection system concerns the vast range of sensors that one has available in order to integrate multiple data. Although, in early vision systems, the fusion of multiple cameras by stereopsis was the best choice, today, laser scanners have brought other alternatives to extract more information about the objects. Existing laser and vision systems, however, still faces the problem of relying on both sensors at the same time. They do not use independence in terms of synergistic sensor detection, but only in terms of lack of relationship between sensor data. This way, we demonstrate that simple reasoning can be beneficial not only when one of the sensors fail, but also to treat the problem of partial segmentation with spatial relationship among the attributes of object parts.

Because object detection systems are inherently human-centered, other aspects of effectiveness with respect to its acceptability are crucial to have a successful application[2]. Although this discussion is out of the scope of our work, an interesting analysis, including also market data, can be found in [Bishop 2005].

7.2 Toward system implementation

As we stated in the previous section, achieving a high detection rate is almost mandatory to embody various sub-systems in a detection architecture. This usually leads to an increase in the computational cost, limiting an on-the-fly system application. In this case, the goal is to find a trade-off between processing time and detection performance, frequently by decreasing the detection accuracy. On the other hand, cutting-edge technology has been coming quickly, causing that trade-off to become more subtle, without having to "bleed" the detector's performance. For instance, MobilEye[3] successfully embedded a vehicle and pedestrian detection system in a special dual-core processor, which equips some of automaker's vehicles, since 2007, such as: Opel, BMW and Volvo. Some research description of that company work can be found in [Reisman 2004], [Shashua 2004], [Stein 2005].

Actually, there has been a tendency for multi-core machines to have not only the cost, but also the size decreased. This fact points to the need of massive parallel algorithms to be designed in order to fullfill all the cores of those hardwares, enabling high performance execution. We have followed this idea by implementing one of our component detectors in a high parallel machine (refer to Appendix C for a description). The other sub-systems of our framework can also be implemented in a parallel version, and will be ported to that multi-core hardware, in the near future.

7.3 Future directions

Temporal information is of great importance not only to help in reducing the detector's false alarm in many circumstances, but also to extract patterns of the object motion.

[2] Following this idea, our business plan of a perception system for carrier trucking won a third place out of 267 participants, in the Intel/GV Entrepreneurship and Venture Capital Business Plan Competition (http://www.cepe.fgvsp.br/desafio/email/resultado.html).

[3] The company site is available in http://www.mobileye.com/

In early detection systems, for instance [Wohler 1998], it was proved to improve the detection rate; on the other hand, by tracking the objects of the scene, a perception system can provide more feedback to the navigation with respect to patterns of future object behavior [Weiming 2006]. Although our detection system has obtained a state-of-the-art performance when classifying objects in difficult situations, temporal analysis could significantly enrich information provided by the perception system. We hope to explore this feature in future work.

Chapter 8

Conclusion

(...) when we have done our utmost to arrive at a reasonable conclusion, we still, when we can reason and investigate no more, must close our minds for the moment with a snap, and act dogmatically on our conclusions.

George Bernard Shaw, in "Preface to Androcles and the Lion"

In this thesis, we have described a vision and laser fusion system built on synergistically combined sub-systems. In the vision space, the key contribution was an ensemble of classifiers which was extensively evaluated over several data sets with different characteristics. This procedure led to an in-depth analysis of the behavior of our vision system with respect to real-life application. In laser space, we contributed with a flexible method able to deal with partial segmentation, and defined on some contextual assertions. Finally, we semantically integrated both sensor data. This fusion was conceived on the spatial relationship among the attributes of the objects after the detection by parts-based classifiers in each sensor space.

Simple reasoning demonstrated effectiveness of the proposed system, following the idea of using contextual information to improve detection performance. Those contextual assertions were mainly object-centered, but the enumeration of global constraints can possibly improve even more the system performance, as demonstrated in recent works. For the target application, that is, perception tasks on low-speed vehicles, the proposed framework demonstrated its validity regarding detection accuracy.

Although we have proceeded one step further concerning the problem of detection time by using a multi-core machine in order to speed up our algorithms, it represented just the beginning of a system design to embed our proposed framework. To jump to a real-life application much has still to be done with respect to processing time, decreasing of FAR (keeping the high HR), and evaluation under different scenarios and weather conditions. These limitations offer, however, interesting researching topics in the future.

Finally, we can state that the key success of a detection system resides on a synergistic combination of sub-systems. These system pieces should provide small specialization of tasks, which together can accomplish the final objective. Although, it could be done with a single sensor, a real-life application demands multiple data sources in order to provide a significant support to system's reliability.

Part IV

Appendices

Appendix A

Support vector machines

SVM was originally proposed by Vapnik [Vapnik 1995] as a binary classifier, which embodies the structural risk minimization (SRM), Vapnik-Chervonenski (VC) dimension theory and optimization theory.

Let $\mathbf{X}$ represent the input space, and Y the desired output space. A set S of training examples, denoted by: $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)) \subseteq (\mathbf{X} \times Y)^l$, where l is the number of training instances. Yet, let $(\mathbf{w}, b)$ denote the weight vector, $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, and b is a bias factor. Then, the hyperplane that separate the input data is given by

$$f(\mathbf{x}) =< \mathbf{w} \cdot \mathbf{x} > +b = \sum_{i=1}^{n} w_i \cdot x_i + b \ . \tag{A.1}$$

An input vector belongs to a certain class if $\text{sign}\,(f(\mathbf{x}))$ is positive, where $\text{sign}\,(f(x))$ is defined by

$$\text{sign}\,(f(\mathbf{x})) = \begin{cases} 1, & \texttt{if } f(\mathbf{x}) \geq 0, \\ -1, & \text{otherwise.} \end{cases}$$

The data is guaranteed to be separable if there exists $(\mathbf{w}, b)$ such that for $i = 1, \ldots, l$

$$< \mathbf{w} \cdot \mathbf{x}_i > +b \geq 1 \qquad \texttt{if} \qquad y_i = 1, \tag{A.2}$$

$$< \mathbf{w} \cdot \mathbf{x}_i > +b < -1 \qquad \texttt{if} \qquad y_i = -1. \tag{A.3}$$

The above equations (A.2), and (A.3), can be combined as

$$y_i \,(< \mathbf{w} \cdot \mathbf{x}_i > +b) \geq 1, \quad i = 1, 2, \ldots, l. \tag{A.4}$$

In order to obtain an optimal separating hyper-plane, let us consider the following *optimization problem*

$$\begin{aligned} \textit{minimize}_{\mathbf{w},b} \qquad & <\mathbf{w}\cdot\mathbf{w}>, && \text{(A.5)} \\ \textit{subject to} \qquad & y_i(<\mathbf{w}\cdot\mathbf{x}_i>+b)\geq 1, \quad i=1,2,\ldots,l. && \text{(A.6)} \end{aligned}$$

In statistical learning theory (SLT) [Vapnik 1995], such optimization problem is quadratic and convex and there is only one *global minimum*. So, by finding a solution to this problem we are also finding the optimal solution.

If the dataset for classification is linearly separable, then the resulting hyperplane of the optimization problem (A.6) separates all the samples of the two classes. Otherwise, some data points of the training set will be misclassified by the solution hyperplane. Such samples are called *outliers*. Also, in general, during the prediction phase, the SVM classifier can misclassify some input data. In this context, an *optimal separating hyperplane* is defined as the hyperplane which minimizes the probability that randomly generated examples are misclassified [Vapnik 1995].

The optimization problem in (A.5)-(A.6) has a solution only if the problem is linearly separable, i.e., only if all the conditions in (A.6) can match. In order to integrate this possibility, *slack variables* ξ_i $(i=1,\ldots,n)$ are introduced into the formulation and subsequent solution of the optimization problem. For outlier samples, slack variables are positive, and represent the distance measured along an axis perpendicular to the separating hyperplane, indicating the amount of "degradation" in the sample classification. Slack variables are zero for non-outlier samples. With the introduction of slack variables, the following minimization problem is obtained

$$\begin{aligned} \textit{minimize}_{\mathbf{w},b} \qquad & <\mathbf{w}\cdot\mathbf{w}>+C\sum_{i=1}^{l}\xi_i, \\ \textit{subject to} \qquad & y_i(<\mathbf{w}\cdot\mathbf{x}_i>+b)\geq 1-\xi_i, \\ & \xi_i\geq 0, \end{aligned} \qquad \text{(A.7)}$$

where C is a large positive constant.

In the optimization problem (A.7), the objective function can be combined with the restrictions. For this propose, the problem is reformulated such that the following Lagrangian function should be minimized

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha, \mathbf{r}) &= \frac{1}{2} < \mathbf{w} \cdot \mathbf{w} > + C \sum_{i=1}^{l} \xi_i \\ &\quad - \sum_{i=1}^{l} \alpha_i \left(\xi_i + y_i \left[(\mathbf{w} \cdot \mathbf{x}_i) + b \right] - 1 \right) - \sum_{i=1}^{l} r_i \xi_i \qquad \text{(A.8)} \\ \textit{subject to} &\quad \alpha_i \geq 0, r_i \geq 0. \qquad \text{(A.9)} \end{aligned}$$

In (A.8)-(A.9) $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a vector of Lagrange multipliers, $\mathbf{r} = (r_1, r_2, \ldots, r_n)$, and $r_i = C - \xi_i$ $(i = 1, \ldots, n)$. This is called the primal Lagrangian formulation. The minimum solution to this primal optimization problem must satisfy the Karush-Kuhn-Tucker conditions [Cristianini 2000]. The solution of the Lagrangian using the primal form is difficult, because it contains inequalities constraints. Hence, an alternative form should be constructed by setting to zero the derivatives of the primal variables, and substituting the relations in the initial Lagrangian, removing the primal variables from the expressions. That casts another optimization problem called "dual".

To find a solution to the dual formulation (A.8)-(A.9), the zeros of the partial derivatives of $L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})$, with respect to the primal variables $\mathbf{w}$, b, and ξ_i $(i = 1, \ldots, n)$ must be obtained

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=0}^{l} y_i \alpha_i \mathbf{x}_i = 0, \qquad \text{(A.10)} \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial \xi_i} &= C - \alpha_i - r_i = 0, \quad i = 1, \ldots, n, \qquad \text{(A.11)} \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha, \mathbf{r})}{\partial b} &= \sum_{i=0}^{l} y_i \alpha_i. \qquad \text{(A.12)} \end{aligned}$$

Substituting the solutions $\mathbf{w}$, ξ_i and b of (A.10)-(A.12) in expression of the primal objective function (A.8), the following dual objective function is obtained

$$L(\alpha, \mathbf{r}) = \sum_{i=0}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=0}^{l} y_i y_j \alpha_i \alpha_j < \mathbf{x}_i \cdot \mathbf{x}_j > . \qquad \text{(A.13)}$$

In that case, the objective function (A.13) is constrained by the following conditions

$$\alpha_i \left[y_i \left(< \mathbf{w} \cdot \mathbf{x} > +b \right) - 1 + \xi \right] = 0, \qquad i = 1, ..., l, \tag{A.14}$$

$$\xi_i \left(\alpha_i - C \right) = 0, \qquad i = 1, ..., l. \tag{A.15}$$

The optimization of the dual problem (A.13)-(A.15) yields the optimal values of α, and $\mathbf{r}$. Then, the optimal value of $\mathbf{w}$ for the primal problem can be found from (A.10) - let us call it $\mathbf{w}^*$. Since b does not appear in the dual problem, its optimal value, b^* must be found from the primal constraints

$$b^* = \frac{\max_{y_i=-1}(< \mathbf{w}^* \cdot \mathbf{x}_i >) + \min_{y_i=1}(< \mathbf{w}^* \cdot \mathbf{x}_i >)}{2}. \tag{A.16}$$

The quadratic optimization problem formulated in (A.13)-(A.15) can be solved using the sequential minimal optimization (SMO) [Platt 1999], or the algorithm found in [Joachims 2002].

The above formulation can be used when data (training and prediction) is assumed to be linearly separable. This is attained if a low error rate is obtained after training. If this does not hold, the above formulation is extended as described below.

The extension from the previous method is done by mapping the input features ($\mathbf{x}_i$) using a non-linear function. This mapping is done by a kernel function, which maps the input features into a higher dimensional space, named *feature space*, such that data is linearly separable after mapping.

For the non-linear mapping, the dual Lagrangian formulation (A.13) is used. This mapping is implicitly performed by substituting the inner product $< \mathbf{x}_i \cdot \mathbf{x}_j >$ by $K(\mathbf{x}_i, \mathbf{x}_j)$ in equation (A.13).

$$< \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) >= K(\mathbf{x}_i, \mathbf{x}_j) \tag{A.17}$$

As a consequence of this mapping approach no additional change is required in the phases of training and prediction. The solution of the problem remains the same as described above. This has another desirable consequence: the computational effort of the method is not significantly affected by the kernel formulation.

A function may be considered a *kernel* if it satisfies Mercer's condition, i.e., the matrix $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$, that represents the kernel function, is positive semidefinite. There are several *kernel* functions proposed in the literature. The most usual kernels are the following:

1. linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ (no mapping).

2. polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + h)^d, \quad \gamma > 0.$

3. Gaussian RBF: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \| \mathbf{x}_i - \mathbf{x}_j \|^2), \quad \gamma > 0.$

4. Multilayer perceptron: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + h).$

where γ, h e d are *kernel* parameters.

The application of the kernel mapping would be equivalent to the substitution of the input feature vectors $\mathbf{x}$ by $\Phi(\mathbf{x})$, i.e., the referred *kernel* mapping can be seen as $\mathbf{x} = (x_1, ..., x_l) \longmapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_l(\mathbf{x}))$. The new set of input features may now be defined as $F = \{\phi(\mathbf{x}) \mid \exists y \in Y : (\mathbf{x}, y) \in S\}$. To take into account the feature set F, an optimization strategy is now applied as if it were a linearly separable space and this linear SVM has the following modified form:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{w}_i \phi(\mathbf{x}) + b \ . \tag{A.18}$$

Finally, it is noteworthy that to decide if the input space is linearly separable, initially, a SVM with linear kernel is tested to see if the resulting separating hyperplane separates the data. If the training error is high or there is no convergence in the solution of the problem, then a different *kernel* function is applied in order to map the input space into a higher dimensional space. After the kernel mapping, the SVM problem is solved as described above.

Appendix B

Classifier fusion methods

Let $V = \{V_1, V_2, V_3\}$ be a set of feature extractor-classifiers, such that $V_i : \mathbb{R}^n \rightarrow \Omega_i$, where $\Omega_i \in \{1, -1\}$, $i = 1, 2, 3$, assigns a binary class label for V_i. Rather than +1 or -1 as output, some classifiers provide a confidence score output (s_c), obtained from the distance between the input vector and the separating hyperplane. This confidence score must be scaled to be used as input in fuzzy systems.

B.1 Majority vote

For MV, sum (S_{MV}), weighted (W_{MV}) and heuristic[1] (H_{MV}) rules are given by

$$S_{MV} = sign(\sum_{i=1}^{3} \Omega_i)\,. \tag{B.1}$$

$$W_{MV} = sign(\sum_{i=1}^{3} b_i \Omega_i)\,, \tag{B.2}$$

where $b_i \propto log\frac{p_i}{1-p_i}$ and p_i is the global accuracy of each component classifier over a training data set.

$$H_{MV} = \begin{cases} 1, & \sum_{i=1}^{3} \Omega_i > 0 \; or \; \Omega_1 = 1 \; or \; \Omega_2 = 1 \\ -1, & \sum_{i=1}^{3} \Omega_i < 0 \; or \; \Omega_3 = -1 \end{cases} \tag{B.3}$$

B.2 Fuzzy integral

Let X be an input vector with N_c elements, where N_c is the number of classifiers, and $\mathcal{P}(X)$ be the power set E. A fuzzy measure $g : \mathcal{P}(X) \rightarrow [0, 1]$ represents the individual importance of each classifier V_i, satisfying the following properties

i) $g(\emptyset) = 0$, $g(X) = 1$ (boundary conditions).

[1] This method was particularly conceived for the NiSIS competition.

ii) $g(A) \leq g(B)$ if $A \subset B$, for any subsets A, $B \in \mathcal{P}(X)$ (monotonicity).

g is called a λ-fuzzy measure for any subsets A, $B \in \mathcal{P}(X)$, and $A \cap B = \emptyset$, such that

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B)\,. \tag{B.4}$$

where $\lambda \geq -1$ denotes the degree of interrelation between subsets A and B, and is according to

$$\lambda + 1 = \prod_{i=1}^{N_c} (1 + \lambda g_i)\,, \tag{B.5}$$

where g_i is a fuzzy measure used to express the decision support for each individual classifier.

Let $h : (X) \rightarrow [0, 1]$ be a membership function, which monotonically decreases with respect to each element of X (if it does not hold, X must be resorted), and be $H = \{h_i, i = 1, ..., N_c\}$, then Sugeno FI is defined as follows:

$$\int h(x) \circ g = \max_{E \subseteq X} \{\min_{x \in E} (h(x), g(E))\}\,. \tag{B.6}$$

The Choquet FI differs in the way it is computed, such that

$$\int h(x) \circ g = h_1(x) + \sum_{i=2}^{2^{N_c}} [h_{i-1}(x) - h_i(x)] g_{i-1}\,. \tag{B.7}$$

Appendix C

Parallel architecture for object detection

A typical system for object recognition in still images is usually composed of three main modules: object searching, object recognition and object tracking. These modules are primarily conceived to be integrated, while information transfer among them should provide more confidence in the decision of what or where the objects in the images are. Yet, the first two modules are probably the most computationally expensive since, after the objects are hypothesized, a Kalman-based, for instance, tracking is almost costless.

To build a complete object detection system for real life applications, some project issues should be considered. The main one resides in obtaining a trade-off between a high performance object detection, and on-the-fly implementation. In ITS field, for instance, applications hold typically 15 fps in order to provide an effective timing system. This frame rate usually decreases while increasing the complexity of the algorithms applied.

In this appendix, we present a framework to parallelize object detection algorithms using cell broadband engine (CBE[1]) inside a PlayStation3 (PS3) platform. At this stage, we aimed to parallelize the HOG detection system. It is important to note that it is worthless to use any parallel hardware if the algorithm is not parallelizable.

C.1 Object searching

Finding objects in still images is generally accomplished by two main methods: brute force (sliding window) or moving object segmentation. In the first category[2], a normalized image window is shifted through various scales and positions over the input frame. The aim of this task is such that, in each position of the normalized window, features are extracted, feeding a trainable classifier, which will decide if the position contains an object or not. Figure C.1 depicts the main idea of that pyramidal searching method. Concerning

[1] This is referred here just as Cell.

[2] That one which we are focused in this work.

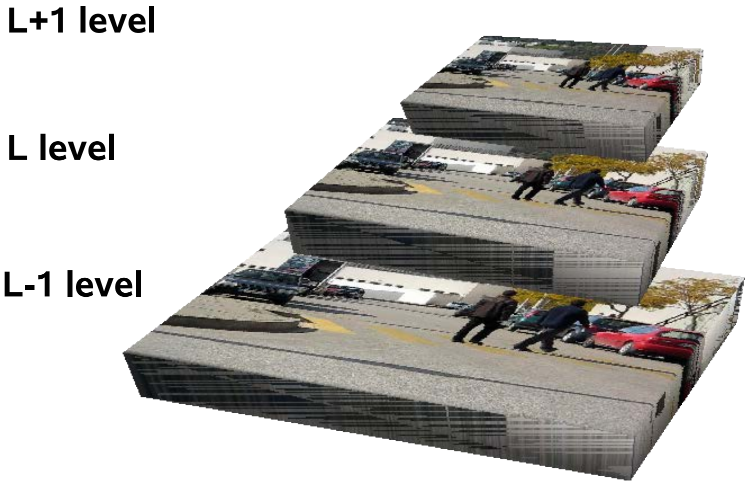


Figure C.1: A normalized image window runs through all scales and positions over the input frame. The smallest is the scale of the input frame, the closest is the object.

the effectiveness of finding an object, these techniques depend only on the performance of the classifier, which will be usually higher as more complex algorithms are used.

Under the perspective of the parallelization onto PS3 platform, brute force methods rely on different computing parallelization strategies:

- Instruction level parallelization in the synergistic processing unit (SPE);
- Image scale level parallelization – each image scale should be sent to an SPE to be processed.

The rationale behind the first strategy resides in speeding up the whole process, parallelizing as much as possible the way that each instruction of the algorithm is performed. The parallelization of the latter method, instead, should happen at the moment that each scale of the pyramid (or part of one image) is sent to each SPE, making the parallelization to be accomplished in a higher level programming technique. Yet, if the size of each scaled image is longer than the Local Store (LS) (see Fig. C.3) presented in each SPE, one should think about other strategies to break the processing in multiple clock cycles.

In moving object segmentation methods, the way that the image ROI is found usually relies on faster methods, since the main objective is to segment what is foreground and what is background based on hints of movement. The success of this type of method resides in a good hypothesis technique based on various clues. In other words, it is better to obtain false positives than miss detections, in view of the fact that in the object

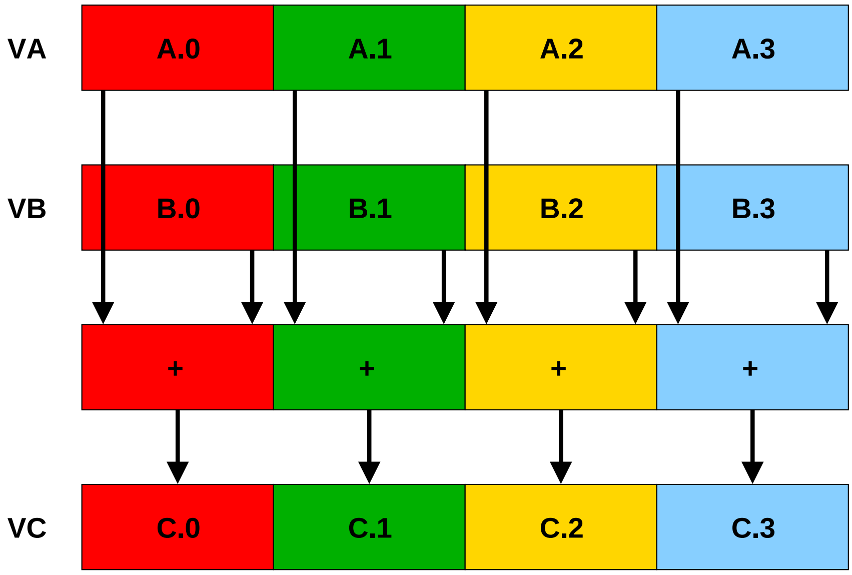


Figure C.2: SIMD processing: Adding two vectors VA and VB, sending the result to a third vector VC [Buttari 2007].

classification step a false positive can be corrected by a more reliable classifier method. The parallelization of these methods depend on the specific algorithms used. Generally, optical flow and particle filter are used to estimate the difference between the foreground and the background. In this way, an instruction level parallelization technique should present better results, since a scale level approach is not usually feasible.

C.1.1 Object classification

In this section, we describe how deterministic classification approaches can be parallelized. This type of classification system fits perfectly the acceleration approach implemented by the Cell, since this architecture is composed of vector processors. The vectorization structure takes place by means of single instruction multiple data (SIMD) instructions, being able to deal with multiple data in the same clock. The programming approach is performed by using C language with extended command to explore the SIMD processor, and vectorization tasks. Fig. C.2 illustrates an example of a vector operation.

Another aspect of optimization is that all methods used in the ensemble rely on deterministic methods. Therefore, the use of SIMD processing can be used, and the parallelism can be achieved not only in the instruction level, but also in the image level.

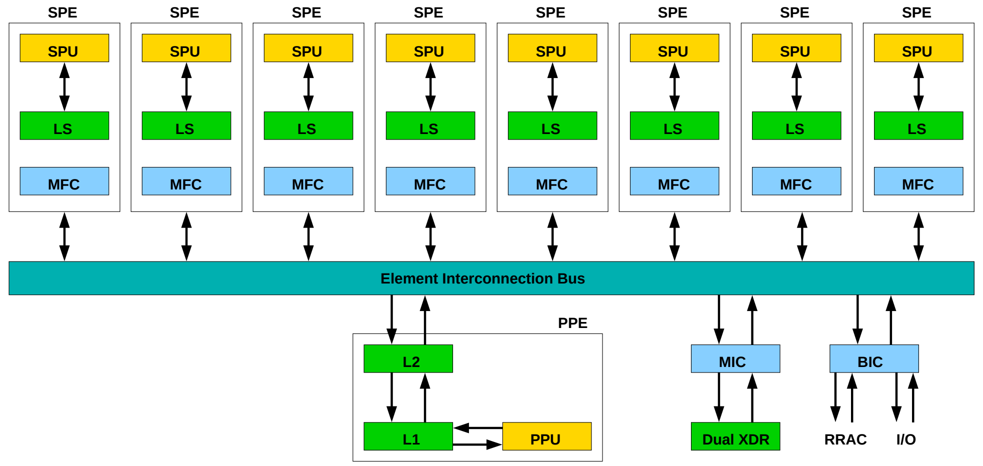


Figure C.3: Cell processor design [Buttari 2007]: The architecture is composed by a dual-threaded processor called PPE and 8 vector processors, called SPE. Only 6 SPEs are programable available under Linux.

C.2 Exploring parallelism using Cell

Cell architecture is a powerful heterogeneous multi-core processor, developed to be used in many platforms, such as high-definition televisions, supercomputers and game consoles, namely the Sony PS3. Nowadays, PS3 is used in many scientific computing projects due to presenting a suitable trade-off between cost and benefits.

This section presents the Cell architecture, as well as a preliminary study about the viability of using the Cell processor (using PS3) in image object detection for ITS.

C.2.1 Cell architecture

Cell is a heterogeneous multi-core architecture, containing a dual-threaded processor unit, denominated power processing element (PPE) and eight SPE co-processors. Figure C.3 depicts Cell architecture.

PPE is a 64-bit, two-way simultaneous multithreading processor which is compliant with the PowerPC 970 Architecture. PPE has one power processor unit (PPU), 32 Kbytes of L1 cache, and 512 Kbytes of L2 cache. PPU uses the PowerPC 970 instruction set with a SIMD engine called vector multimedia extension (VMX). SPEs, where the real power of Cell processor is, consist of a synergistic processing unit (SPU), 256 Kbytes of LS, and a memory flow controller (MFC) which delivers powerful direct memory-access capabilities to the SPU using direct memory address (DMA). The SPEs has a 128-bit vector register file and a range of SIMD instructions that can operate simultaneously on two double-

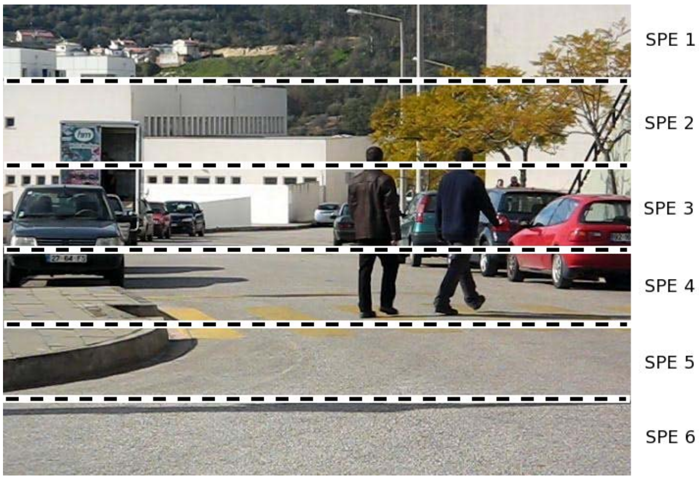


Figure C.4: Fragmentation approach: A single image is fragmented into same sized pieces. Each piece is then sent to one different SPE.

precision values, four single-precision values, eight 16-bit integers or 16 8-bit characters.

There are instructions able to complete a vector operation in one clock cycle. The Cell components are connected via the element interconnection bus (EIB). EIB has four unidirectional rings, two in each direction. Additionally, there is a token-based mechanism which permits a internal speed of 204.8 Gbytes/s.

The Cell is mainly a distributed memory processor, where each SPE has its private memory. There is an explicit control over data motion, and one can use mechanism like mailbox to exchange data messages.

C.2.2 Optimizing object detection algorithms

Object detection algorithms seem to be suitable for a Cell optimized execution. At this stage, we identify two approaches to use the whole power of Cell processor in order to classify objects in images: fragmentation of the image in equal number of pieces (sending each piece to a different SPE), and a pyramidal searching. The first approach is illustrated in Fig. C.4.

The goal is to split the image into six pieces of the same size. After that, each SPE receives one of these pieces and executes the object detection algorithm over the image piece. At the same time, each image piece is processed by each SPE, improving the performance of the algorithm execution. The PPE has the role of fragmenting the original image, sending the resulting pieces to the SPEs.

The second approach (Fig. C.5) uses the multi-resolution technique to create a pyra-

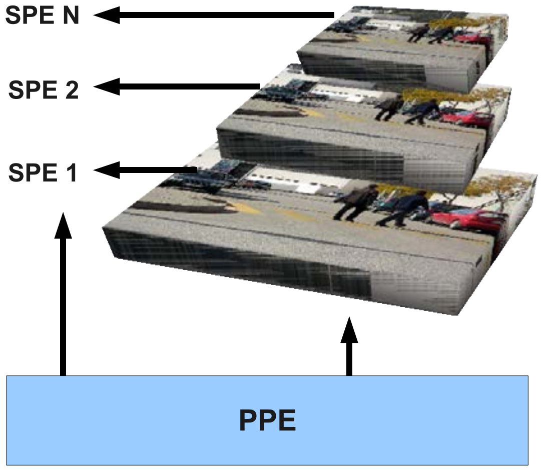


Figure C.5: Multi-resolution approach: Each level of the pyramid is sent to an SPE. This approach is feasible as long as each scaled image can fit into the LS of each SPE. Otherwise, a multiple step technique should be implemented to process each image, considering multiple references to the LS.

mid of six stages, where each stage is the original image in a different resolution. Each SPE receives one pyramid level, performing the object detection algorithm on it. PPE is in charge of receiving each frame, scaling it, and finally sending the results to each SPE.

C.2.2.1 HOG optimization on Cell

To evaluate the potential of the Cell platform, we implemented the multi-resolution approach over a sliding window searching method with a HOG detector. Figure C.6 illustrates the architecture of the system.

The system is comprised of 3 modules, with modules (1) and (3) being run in PPU, while module (2) runs in the SPUs. Frames are provided offline, that is, there is no camera acquisition. To speed up the process, gradients and magnitudes are computed for all the octaves (scaled images) of the input image. Since there is a bottleneck in the EIB to transfer the data from the main memory to the LS of each SPU, each octave is broken into small pieces of 16Kbytes each. Then, at one interaction of the algorithm in SPU, one block of gradient and one block of magnitude are used to computing HOG features, settling a transfer matrix to be sent to each LS. That is all made in parallel. After processing all the blocks, the system copies the array of detected objects to the main memory. At the end, the results are shown through a double framebuffer, which accelerates the video information processing.

With this parallel architecture and without any other vectorization instruction, we

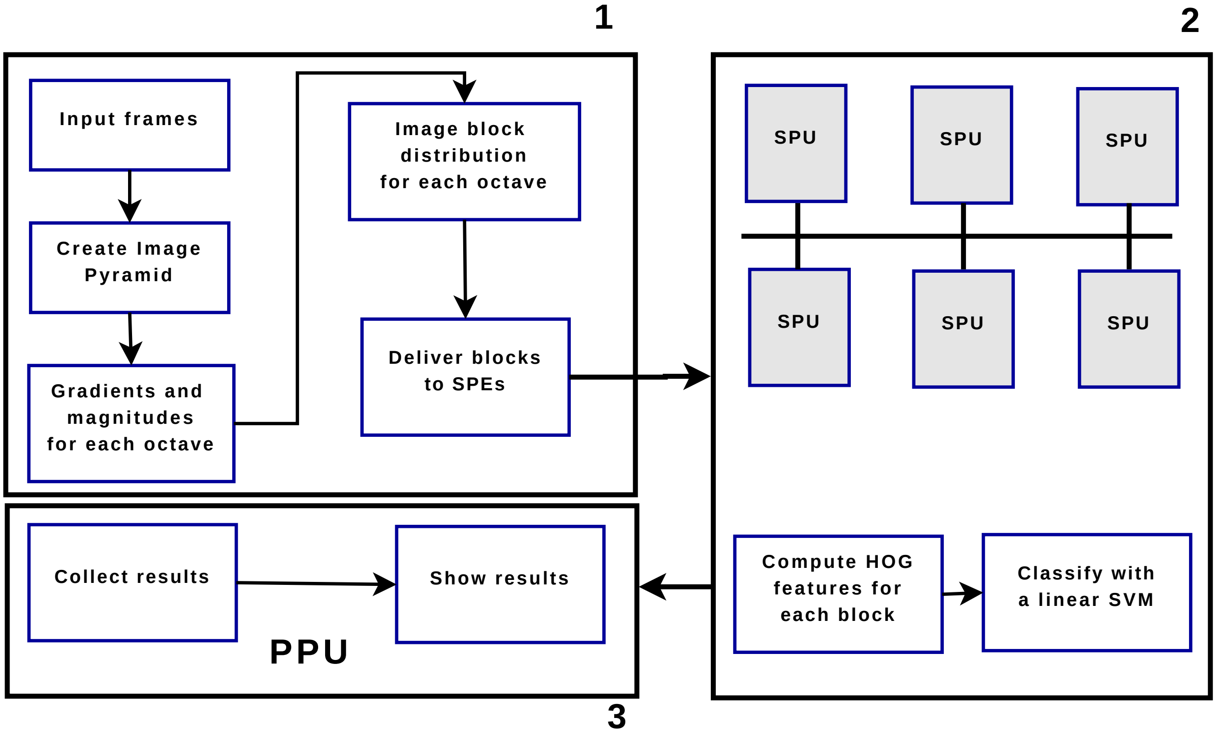

1
2
Input frames
Create Image Pyramid
Gradients and magnitudes for each octave
Image block distribution for each octave
Deliver blocks to SPEs
SPU
SPU
SPU
SPU
SPU
SPU
Collect results
Show results
PPU
3
Compute HOG features for each block
Classify with a linear SVM

Figure C.6: Framework of the parallel HOG. This is comprised of 3 modules: In module (1), the input image is broken into n octaves (scaled input image); for each octave, the image is broken into smaller pieces to be sent to the SPUs. In module (2), the smaller pieces of an image come in parallel, while in each SPU a piece of code is responsible for extracting HOG features, posteriorly classifying them with a linear SVM. Finally, in module (3), the results are shown. Modules (1) and (3) run in PPU, while module (2) runs in SPUs.

were able to decrease the processing time of a 640 × 480 input image by 4 times, going from 36 secs down to 9 secs, in comparison to a dual-core machine.

References

[Aksela 2006] M. Aksela and J. Laaksonen. *Using diversity of errors for selecting members of a committee classifier.* Journal of Pattern Recognition, vol. 39, pages 608–623, 2006.

[An 2009] S. An, P. Peursum, W. Liu and S. Venkatesh. *Efficient algorithms for subwindow search in object detection and localization.* In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, pages 264–271, 2009.

[Andriluka 2008] M. Andriluka, Roth. S. and B. Schiele. *People-tracking-by-detection and people-detection-by-tracking.* In Proc. of IEEE Intl. Conference on Computer Vision and Pattern Recognition, 2008.

[Araujo 2008] R. Araujo, U. Nunes, L. Oliveira, P. Sousa and P. Peixoto. Support vector machines and features for environment perception in mobile robotics, pages 219–250. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2008.

[Arras 2007] K. Arras, O. Mozos and W. Burgard. *Using boosted features for the detection of people in 2D range data.* In Proc. of IEEE Intl. Conf. on Robotics and Automation, pages 3402–3407, 2007.

[Bishop 2005] R. Bishop. Intelligent vehicle tecnology and trends. Artech House, 2005.

[Bishop 2006] C Bishop. Pattern recognition and machine learning. Springer, 2006.

[Borges 2000] A. Borges and M. Aldon. *A split-and-merge segmentation algorithm for line extraction in 2D range images.* In Proc. of Intl. Conf. on Pattern Recognition, volume 1, pages 441–444, 2000.

[Broggi 2008] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri and J. Gi. *Localization and analysis of critical areas in urban scenarios.* In Proc. of IEEE Intl. Symp. on Intelligent Vehicles, pages 1074–1079, 2008.

[Buttari 2007] A. Buttari, P. Luszczek, J. Kurzak, J. Dongarra and G. Bosilca. *A rough guide to scientific computing on the PlayStation 3.* Rapport technique, 2007. Technical Report UT-CS-07-595.

[Choquet 1954] G. Choquet. *Theory of capacities.* In Annales de l'Institut Fourier, volume 5, pages 131–195, 1954.

[Cristianini 2000] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machine and other kernel-based learning methods. Cambridge University Press, 2000.

[Dalal 2005] N. Dalal and B. Triggs. *Histograms of oriented gradients for human detection.* In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, volume 1, pages 886–893, 2005.

[Dalal 2006] D. Dalal. *Finding People in Images and Videos*. PhD thesis, Institut National Polytechnique de Grenoble, 2006.

[Dollar 2009] P. Dollar, C. Wojek, B. Schiele and P. Perona. *Pedestrian detection: A benchmark*. In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, pages 304–311, 2009.

[Douillard 2007] B. Douillard, D. Fox and F. Ramos. *A spatio-temporal probabilistic model for multi-sensor object recognition*. In Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pages 2402–2408, 2007.

[Dryden 1998] I. Dryden and K. Mardia. Statistical shape analysis. Wiley, 1998.

[Forstner 1999] W. Forstner and B. Moonen. *A metric for covariance matrices*. Rapport technique, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[Freund 1996] Y. Freund and R. Schapire. *Experiments with a new boosting algorithm*. In Intl. Conf. on Machine Learning, pages 148–156, 1996.

[Gandhi 2007] T. Gandhi and M. Trivedi. *Pedestrian protection systems: Issues, survey, and challenges*. IEEE Trans. on Intelligent Transportation Systems, vol. 8, pages 413–430, 2007.

[Garcia 2004] C. Garcia and M. Delakis. *Convolutional face finder: A neural architecture for fast and robust face detection*. Proceedings of the IEEE, vol. 26, no. 11, pages 1408–1423, 2004.

[Haykin 2008] S. Haykin. Neural networks and learning machines. Prentice Hall, 2008.

[Hoiem 2006] D. Hoiem, A. Efros and M. Hebert. *Putting objects in perspective*. In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, pages 2137–2144, 2006.

[Joachims 1962] T. Joachims. *Hurley, F. and Cattel, R.* In The Procrustes program: Producing direct rotation to test a hypothesized factor structure, volume 7, pages 258–262, 1962.

[Joachims 2002] T. Joachims. *Optimizing search engines using clickthrough data*. In Proceedings of the ACM Conf. on Knowledge Discovery and Data Mining, pages 133–142, 2002.

[Kecman 2001] V. Kecman. Learning and soft computing: Support vector machines, neural networks, and fuzzy logic models. 2001.

[Kirkpatrick 1985] D. Kirkpatrick and J. Radke. Computational geometry, chapitre A framework for computational morfology. North-Holland Publishing Company, 1985.

[Kuncheva 2004] L. Kuncheva. Combining pattern classifiers: Methods and algorithms. Wiley-Interscience, 2004.

[Lambert 2006] D. Lambert. *Formal Theories for Semantic Fusion.* In Proc. of Intl. Conference on Information Fusion, pages 1–8, 2006.

[Lampert 2007] H. Lampert, M. Blaschko and T. Hofmann. *Dynamic 3D scene analysis from a moving vehicle.* In Beyond Sliding Windows: Object localization by rfficient subwindow search, volume 1, pages 1–8, 2007.

[Lawrence 1997] S. Lawrence, L. Giles, A. Tsoi and A. Back. *Face recognition: A convolutional neural network approach.* IEEE Transactions on Neural Networks, vol. 8, no. 1, pages 98–113, 1997.

[LeCun 1995] Y. LeCun and Y. Bengio. The handbook of brain theory and neural networks, chapitre Convolutional networks for images, speech and time series. MIT Press, 1995.

[Lecun 1998] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. *Gradient-based learning applied to document recognition.* vol. 86, no. 11, pages 2279–2324, 1998.

[Leibe 2005] P. Leibe, E. Seemann and B. Schiele. *Pedestrian detection in crowded scenes.* In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, volume 1, pages 878–885. IEEE, 2005.

[Leibe 2007] B. Leibe, N. Cornelis, K. Cornelis and L. Van Gool. *Dynamic 3D scene analysis from a moving vehicle.* In Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition, volume 1, pages 1–8, 2007.

[Llorca 2006] L. Llorca, M. Sotelo, L. Bergasa, P. Toro, J. Nuevo, M. Ocana and M. Garrido. *Combination of feature extraction methods for SVM pedestrian detection.* IEEE Transactions on Intelligent Transportation Systems, vol. 8, pages 292–307, 2006.

[Lowe 1999] D. Lowe. *Object recognition from local scale-invariant features.* In Proc. of IEEE Intl. Conf. on Computer Vision, volume 2, pages 1150–1157, 1999.

[Lowe 2004] D. Lowe. *Distinctive image features from scale-invariant keypoints.* International Journal of Computer Vision, vol. 60, pages 91–110, 2004.

[Mahlisch 2006] R. Schweiger Mahlisch, W. Ritter and K. Dietmayer. *Sensorfusion using spatio-temporal aligned video and LIDAR for improved vehicle detection.* In Proc. of IEEE Intl. Symp. on Intelligent Vehicles, pages 424–429, 2006.

[Marchette 2004] D. Marchette. Random graphs for statistical pattern detection. Wiley, 2004.

[Munder 2006] S. Munder and D. Gavrila. *An experimental study on pedestrian classification.* In IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 28, pages 1863–1868, 2006.

[Nanni 2008] L. Nanni and A. Lumini. *Ensemble of multiple pedestrian representations.* IEEE Transactions on Intelligent Transportation Systems, vol. 9, pages 365–369, 2008.

[Oliveira 2007a] L. Oliveira, G. Monteiro, P. Peixoto and U. Nunes. *Towards a robust vision-based obstacle perception with classifier fusion in cybercars.* In Proc. of Computer Aided System Theory, Lectures Notes in Computer Science, volume 4739, pages 1089–1096. Springer-Verlag, 2007.

[Oliveira 2007b] L. Oliveira, P. Peixoto and U. Nunes. *Scheme of primate's visual cortex cells for pedestrian recognition*, 2007. Best accuracy model out of 17 participants over Daimler Chrysler image dataset in NiSIS Competition.

[Oliveira 2007c] Luciano Oliveira, P. Peixoto and U. Nunes. *A hierarchical fuzzy integration of local and global feature-based classifiers to recognize objects in autonomous vehicles.* In Proc. of IEEE Intl. Conf. on Robotic Automation Workshop on Planning, Perception and Navigation for Intelligent Vehicles, pages 45–50, 2007.

[Oliveira 2008a] L. Oliveira, R. Britto and U. Nunes. *On Using Cell Broadband Engine for Object Detection in ITS.* In Proc. of IEEE Intl. Conf. on Intelligent Robots Systems 2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles, pages 54–58, 2008.

[Oliveira 2008b] L. Oliveira and U. Nunes. *On integration of features and classifiers for robust vehicle detection.* In Proc. IEEE Intl. Conference on Intelligent Transportation Systems, pages 414–419, 2008.

[Oliveira 2010a] L. Oliveira and U. Nunes. *Context-aware pedestrian detection using LIDAR.* In Proc. IEEE Intl. Intelligent Vehicles Symposium, pages 773–778, 2010.

[Oliveira 2010b] L. Oliveira, U. Nunes and P. Peixoto. *On exploration of classifier ensemble synergism in pedestrian detection.* IEEE Transactions on Intelligent Transportation Systems, pages 16–27, 2010.

[Oliveira 2010c] L. Oliveira, U. Nunes, P. Peixoto, M. Silva and F. Moita. *Semantic fusion of laser and vision in pedestrian detection.* Pattern Recognition, vol. 43, pages 3648–3659, 2010.

[Papageorgiou 2000] C. Papageorgiou and T. Poggio. *A trainable system for object detection.* Intl. Journal of Computer Vision, vol. 38, pages 15–33, 2000.

[Parag 2005] S. Parag and P. Domingos. *Discriminative training of Markov logic networks.* In Proc. Natl. Conf. on Artificial Intelligence, volume 2, pages 868–873, 2005.

[Platt 1999] J. Platt. *Using sparseness and analytic qp to speed training of support vector machines.* In Proc. of Advances in Neural Information Processing Systems, pages 557–563, 1999.

[Platt 2000] J. Platt. *Probabilistic outputs for support vector machines and comparison to regularize likelihood methods.* In A. Smola, P. Bartlett, B. Schoelkopf and D. Schurmans, editeurs, Advances in Large Margin Classifiers, pages 61–74, 2000.

[Poon 2006] H. Poon and P. Domingos. *Sound and efficient inference with probabilistic and deterministic dependencies.* In Proc. Natl. Conf. on Artificial Intelligence, pages 458–463, 2006.

[Porikli 2006] F. Porikli and T. Kocak. *Robust license plate detection using covariance descriptor in a neural network framework.* Proc. of IEEE Conference on Advanced Video and Signal Based Surveillance, pages 107–112, 2006.

[Premebida 2005] C. Premebida and U. Nunes. *Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications.* In Proc. of Natl. Festival of Robotics Scientific Meeting (ROBOTICA), pages 17–25, 2005.

[Premebida 2006] C. Premebida and U. Nunes. *A multi-target tracking and GMM-classifier for intelligent vehicles.* In Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems, pages 313–318, 2006.

[Premebida 2007] C. Premebida, G. Monteiro, U. Nunes and P. Peixoto. *A LIDAR and vision-based approach for pedestrian and vehicle detection and tracking.* In Proc. of IEEE Intl. Conf. on Intelligent Transportation Systems, pages 1044–1049, 2007.

[Rahman 1996] Z. Rahman, D. Jobson and G. Woodell. *Multi-scale retinex for color image enhancement.* In Proc. of IEEE Intl. Conf. on Image Processing, volume 3, pages 1003–1006, 1996.

[Reisman 2004] P. Reisman, O. Mano, S. Avidan and A. Shashua. *Crowd detection in video sequences.* In Proc. of IEEE Intl. Symp. on Symposium on Intelligent Vehicles, pages 66–71. IEEE, 2004.

[Richardson 2006] M. Richardson and P. Domingos. *Markov Logic Networks.* vol. 62, pages 107–136, 2006.

[Ridder 2003] D. Ridder, R. Duin, M. Egmont-Petersen, L. van Vliet and P. Verbeek. *Nonlinear image processing using artificial neural networks.* vol. 126, pages 352–450, 2003.

[Shashua 2004] A. Shashua, Y. Gdalyahu and G. Hayun. *Pedestrian detection for driving assistance systems: Single-frame classification and system level performance.* In Proc. of IEEE Intl. Symp. on Symposium on Intelligent Vehicles, pages 1–6, 2004.

[SICK 2006] SICK. *Laser measurement systems to LMS200 to LMS291*, 2006.

[Song 2002] Z. Song, Y. Chen, L. Ma and Y. Chung. *Some sensing and perception techniques for an omnidirectional ground vehicle with a laser scanner.* In Proc. of IEEE Intl. Symp. on Intelligent Control, pages 690–695, 2002.

[Stein 2005] P. Stein, E. Rushinek, G. Hayun and A. Shashua. *A computer vision system on a chip: A case study from the automotive domain.* In Proc. of IEEE Intl. Conference on Computer Vision and Pattern Recognition, CVPR Workshops, pages 130–134, 2005.

[Sugeno 1974] M. Sugeno. *Theory of fuzzy integrals and its applications.* PhD thesis, Tokyo Institute of Technology, 1974.

[Szarvas 2006] M. Szarvas, U. Sakai and J. Ogata. *Real-time pedestrian detection using LIDAR and convolutional neural networks.* In Proc. of IEEE Intl. Symp. on Intelligent Vehicles, pages 213–218, 2006.

[Torralba 2003] A. Torralba. *Contextual priming for object recognition.* Intl. Journal of Computer Vision, vol. 53, pages 169–191, 2003.

[Tuzel 2008] O. Tuzel, F. Porikli and P. Meer. *Pedestrian detection via classification on Riemannian manifolds.* IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 30, pages 1713–1727, 2008.

[Vapnik 1995] V. Vapnik. The nature of statistical learning theory. Springer-Verlag, 1995.

[Viola 2001] P. Viola and M. Jones. *Rapid object detection using a boosted cascade.* In Proc. IEEE Intl. Conf. on Computer Vision and Pattern Recognition, pages 511–518, 2001.

[Weiming 2006] H. Weiming, X. Xuejuan, F. Zhouyu, D. Xie, T. Tieniu and S. Maybank. *A system for learning statistical motion patterns.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 9, pages 1450–1464, 2006.

[Wohler 1998] C. Wohler, J. Aulanf, T. Portner and U. Franke. *A time delay neural network algorithm for real-time pedestrian recognition.* In Proc. of IEEE Intl. Conf. on Intelligent Vehicles, pages 247–252, 1998.

[Xavier 2005] J. Xavier, M. Pacheco, D. Castro, A. Ruano and U. Nunes. *Fast line, arc/circle and leg detection from laser scan data in a player driver.* In Proc. of IEEE Intl. Conf. on Robotics and Automation, pages 3930–3935, 2005.

[Yilmaz 2006] A. Yilmaz, O. Javed and M. Shah. *Object tracking: A survey.* ACM Comput. Surv., vol. 38, no. 4, 2006.

[Zelditch 2004] M. Zelditch, D. Swiderski, D. Sheets and W. Fink. Geometric morphometrics for biologists: A primer. Elsevier Academic Press, 2004.

[Zenobi 2001] G. Zenobi and P. Cunningham. *Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error.* In Proc. of European Conference on Machine Learning, pages 576–587, 2001.

[Zhang 2003] S. Zhang, M. Adams, F. Tang and L. Xie. *Geometrical feature extraction using 2D range scanner.* In Proc. of IEEE Intl. Conf. on Control and Automation, pages 901–905, 2003.

[Zhang 2004] Q. Zhang and R. Pless. *Extrinsic calibration of a camera and laser range finder (improves camera calibration).* In Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, volume 3, pages 2301–2306, 2004.