Universitat
Autònoma
de Barcelona

# A Global Approach to Vision-Based Pedestrian Detection for Advanced Driver Assistance Systems

A dissertation submitted by **David Gerónimo Gómez** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor en Informàtica**.

Bellaterra, 10 de desembre de 2009

Director

**Dr. Antonio M. López Peña**
Dept. Ciències de la Computació & Computer Vision Center
Universitat Autònoma de Barcelona

Thesis
Committee

**Dr. Krystian Mikolajczyk**
Dept. of Electronic Engineering
University of Surrey

**Dr. Jaume Amores Llopis**
Computer Vision Center
Universitat Autònoma de Barcelona

**Dr. Dariu M. Gavrila**
Faculty of Science, Universiteit van Amsterdam
and Daimler AG Research & Development

**Dr. Oriol Pujol Vila**
Dept. de Matemàtica Aplicada i Anàlisi
Universitat de Barcelona

**Dr. Felipe Lumbreras Ruíz**
Dept. Ciències de la Computació & Computer Vision Center
Universitat Autònoma de Barcelona

European
Mention
Evaluators

**Dr. Raúl Rojas**
Dept. of Mathematics and Computer Science
Freie Universität Berlin

**Dr. Frédéric Lerasle**
Laboratoire d'Analyse et d'Architecture des Systèmes
Université de Toulouse

**Centre de Visió
per Computador**

This document was typeset by the author using LaTeX $2_{\varepsilon}$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

ISBN 978–84–936529–5–1
Printed by Ediciones Gráficas Rey, S.L.

A mi padre.

*The world we have made, as a result of the level of thinking*
*we have done thus far, creates problems we cannot solve*
*at the same level of thinking at which we created them.*

Albert Einstein (1879–1955)

# Acknowledgements

Making a PhD thesis is without doubt the hardest and most tiring project I have ever made. Up to the very moment of the dissertation, from my view as a researcher the thesis book not only represents an [incomplete] compilation of the acquired and proposed knowledge along the last few years. The book pages represent hard work, new friends, a revitalized open mind, and a new and profound admiration for people I had never even heard of. I would like to dedicate some lines to the people and institutions that have supported me along these years.

First, to the Computer Vision Center, the Universitat Autònoma de Barcelona, the Ministry of Education and Science and European Social Fund grant BES-2005-8864 under projects TRA2004-06702 / AUT and TRA2007-62526 / AUT, and Consolider Ingenio 2010: MIPRCV (CSD200700018) for their funding support.

I would like to thank the members of the tribunal, Dr. K. Mikolajczyk, Dr. D.M. Gavrila, Dr. J. Amores, Dr. J. Pujol and Dr. F. Lumbreras, and the evaluators of the thesis Dr. F. Lerasle and Dr. R. Rojas. I also want to thank the anonymous conference and journal reviewers for their invaluable comments along these years.

To all the friends at the Centre for Vision, Speech and Signal Processing of the University of Surrey: H. Uemura, Dr. X. Zou, Dr. C. Ho Chan, B. Goswami, J. Cabello, Y. Zhou and Z. Kalal. Also thanks for such an enjoyable stay to Duncan, Brian and Mariah. My sincere thanks to the Royal Surrey County Hospital staff.

To all the fellows of the Computer Vision Center: D. Aldavert, Dr. A. Lapedriza, Dr. P. Baiget, Dr. X. Baró, Dr. M. Rusiñol, Dr. O. Ramos, Dr. M. Ferrer, I. Huerta, Dr. A. Borras, S. Segui, J. Vázquez, Dr. A. Hernández, Dr. J. Villanueva, Dr. J. Lladós, Dr. M. Vanrell, Dr. E. Valveny, Dr. X. Roca, Dr. B. Raducanu, Dr. F. Vilariño and the ones that for sure I forget. Also my gratitude to the technical staff, R. Gómez and J. Masoliver, and to the administration, M. Merino, A. Espina, A. García, M. Culleré, P. Villa, H. Piulachs, R. Rionegro, M. Martín and G. Kohatsu.

To my closest friends Raúl, Òscar, Jordi and Jose for the *resopons*, the absurd jokes, the chats, the good and bad moments, and specially for being there even when I was not there.

To the ADAS group. Specially to Dr. D. Ponsa, for all his help, code, and advices from the very beginning of the thesis. To Dr. J. Serrat, for introducing me to the driver assistance systems through my bachelor's project, and also for the hurries in the development of the probabilistic 3D algorithm. To the rest of the group, who have also taken part in this work in one way or another: J.M. Álvarez, H. Caballero, D. Vázquez, J. Marín, X. Boix, Dr. F. Lumbreras, Dr. C. Julià, F. Diego, D. Cheda, M. Rouhani and L. Gallo.

To Dr. K. Mikolajczyk, for welcoming me with open arms and for teaching me not only scientific knowledge but also scientific attitude during my stay in Guildford.

To Dr. A.D. Sappa, who deserves special acknowledgements for the huge amount of hours spent in this thesis, the neverending trigonometry problems, the road estimation technique, the sequences recording, etc.

To Dr. A.M. López, my thesis advisor, for believing that I could be a competent coder and researcher. My most sincere gratitude for your patience, advice, wisdom, and for giving me calm when I needed it.

Thanks to my parents Ángel and Núria, who have always encouraged and supported all my projects, and for educating me in curiosity and sacrifice, my most preciated values. Also to all my family: Sara, Domi, Olga, my grandparents and my family-in-law for caring me and giving me a hand whenever I needed it.

The last but well deserved acknowledgement goes to my loved Verónica, the person who has suffered most the good and bad moments along these years. Thanks for your love, understanding, patience and support all along this time.

# Abstract

At the beginning of the 21th century, traffic accidents have become a major problem not only for developed countries but also for emerging ones. As in other scientific areas in which Artificial Intelligence is becoming a key actor, advanced driver assistance systems, and concretely pedestrian protection systems based on Computer Vision, are becoming a strong topic of research aimed at improving the safety of pedestrians. However, the challenge is of considerable complexity due to the varying appearance of humans (e.g., clothes, size, aspect ratio, shape, etc.), the dynamic nature of on-board systems and the unstructured moving environments that urban scenarios represent. In addition, the required performance is demanding both in terms of computational time and detection rates. In this thesis, instead of focusing on improving specific tasks as it is frequent in the literature, we present a global approach to the problem. Such a global overview starts by the proposal of a generic architecture to be used as a framework both to review the literature and to organize the studied techniques along the thesis. We then focus the research on tasks such as foreground segmentation, object classification and refinement following a general viewpoint and exploring aspects that are not usually analyzed. In order to perform the experiments, we also present a novel pedestrian dataset that consists of three subsets, each one addressed to the evaluation of a different specific task in the system. The results presented in this thesis not only end with a proposal of a pedestrian detection system but also go one step beyond by pointing out new insights, formalizing existing and proposed algorithms, introducing new techniques and evaluating their performance, which we hope will provide new foundations for future research in the area.

# Resum

A començaments del segle XXI, els accidents de tràfic han esdevingut un greu problema no només pels països desenvolupats sino també pels emergents. Com en altres àrees científiques on la Intel·ligència Artificial s'ha transformat en un actor principal, els sistemes avançats d'assistència al conductor, i concretament els sistemes de protecció de vianants basats en Visió per Computador, han esdevingut una important línia d'investigació adressada a millorar la seguretat dels vianants. Tanmateix, el repte és d'una complexitat considerable donada la variabilitat dels humans (p.e., roba, mida, relació d'aspecte, forma, etc.), la naturalesa dinàmica dels sistemes d'abord i els entorns no estructurats en moviment que representen els escenaris urbans. A més, els requeriments de rendiment son rigorosos en termes de cost computacional i d'indexos de detecció. En aquesta tesi, en comptes de centrar-nos en millorar tasques específiques com sol ser freqüent a la literatura, presentem una aproximació global al problema. Aquesta visió global comença per la proposta d'una arquitectura genèrica pensada per a ser utilitzada com a marc tant per a la revisió de la literatura com per a organitzar les tècniques estudiades al llarg de la tesi. A continuació enfoquem la recerca en tasques com la segmentació dels objectes en primer pla, la classificació d'objectes i el refinament tot seguint una visió general i explorant aspectes que normalment no son analitzats. A l'hora de fer els experiments, també presentem una nova base de dades que consisteix en tres subconjunts, cadascun adressat a l'avaluació de les diferents tasques del sistema. Els resultats presentats en aquesta tesi no només finalitzen amb la proposta d'un sistema de detecció de vianants sino que van un pas més enllà indicant noves idees, formalitzant algoritmes proposats i ja existents, introduïnt noves tècniques i evaluant el seu rendiment, el qual esperem que aporti nous fonaments per a la futura investigació en aquesta àrea.

# Contents

# Chapter 1

# Introduction

Intelligent machines have been engineered by humans since the appearance of early civilizations. For example, the first programmable machines have been dated back in Ancient Greece in the 1st century BCE [140]. It is said that human-sized automata were built in ancient China in the previous centuries [118]. Furthermore, from the Renaissance until the 20th century, machines like calculators and chess-playing automata were increasingly researched [106]. The formalization of Artificial Intelligence around the 1950s brought intelligent machines to a new dimension, in which their role in human lives has progressively gained importance. At this moment, humans are assisted everywhere: from hazard alarms, medical technology, communications, transportation, etc. The research in this thesis is focused on the role of Computer Vision for driver assistance, which not only represents a hot research topic nowadays but also is of crucial importance for human societies, as it is argued along this chapter.

## 1.1 Advanced driver assistance systems

Automobiles represent one of the key technologies for human development in the modern era. Since their popularization during the 20th century, automobiles have changed societies in many aspects: demographic distribution, urbanism, social interactions, industry growth, environmental alterations, economy development, etc. Moreover, their potential to provide independent, flexible and fast movement to people has lead to new trends in city planning, traveling and employment. According to [121], around 50 million passenger cars and 20 million commercial vehicles are being produced worldwide every year. At this rate, in the next years the number of automobiles in the world will reach one billion units, specially due to emerging economies like India and China.

Unfortunately, together with the many benefits, such a technology has also carried a dark side since the very beginning: traffic accidents. The first death by a motor vehicle was registered in Ireland on 31st August, 1869 [48]. Obviously, in 19th century the number of existing automobiles was low so fatalities were rare. Nowadays, according to the World Health Organization, road accidents represent the 6th cause of death in high-income countries and the 11th worldwide [127, 8]. Every year almost 1.2 million people are killed in traffic crashes while the number of injured rises to 50 million (Figure 1.1). Furthermore, these numbers are expected to increase a 65% between 2000 and 2020, specially in low and middle-income countries.

**Figure 1.1:** In the last years traffic authorities make use of information panels in highways as a dissuasive measure for drivers to speed down, not use mobile phones, etc. As an example, this panel panel in a Spanish highway states the accumulated number of deaths and injuries in Catalonia (Photo by V. Marchán, August 2008).

In order to improve safety, the automobile industry has been developing special technology of increasing complexity and performance through the time. First electric headlamps were introduced in 1898 in the Columbia Electric Car, while turn signal lights were devised in 1907. In the 1920s, physicians advocated the use of seat belts in cars to protect vehicle passengers, but it was not until 1955 when Ford included them as an optional equipment and 1958 when Saab made them standard. Later, Volvo patented the well-known three-points belt, the one used at present time. Safety cage and padded dashboard were incorporated in 1949. Airbag was developed by various researchers in late 1960s, although it would take two decades to become standard in the United States and three decades in Europe. In 1978, Bosch and Mercedes-Benz commercialized the antilock braking system (ABS) technology on trucks and sedans, although the technology had been first engineered for aircrafts, like seat belts. Electronic stability control (ESC) was first introduced by Mercedes-Benz in 1995 (co-developed with Bosch and trademarked as electronic stability programme) followed by BMW and Volvo. This technology provides automatic individual wheel braking when it detects vehicle skids. As can be seen, since the invention of vehicle and until 1990s the technological advances in security relied mostly in physical devices focused on providing safety when accidents were happening.

**In the last twenty years, research has moved toward intelligent systems able to predict dangerous situations and anticipate the accidents. They are referred as advanced driver assistance systems (ADAS), in the sense that**

**they help the driver by providing warnings, assisting to take decisions and even taking automatic evasive actions in extreme cases.** They differ from the previous safety technologies in the sense that they do not only can rely on physical/mechanical cues from the host vehicle (e.g., ABS or ESC) but in addition they understand the exterior world up to some extent. As will be devised during this thesis, Artificial Intelligence plays a key role when pursuing this understanding of the vehicle surroundings. In addition, efforts in other research areas such as sensors, human machine interfaces, and even aspects like psychology and law have to be made when developing these systems.

The first stone in the area of ADAS was put by E. Dickmanns group in 1986 with an autonomous highway driving system [39, 160]. They presented a system able to drive through closed highways at speeds of up to 96 km/h by exploiting cameras, rudimentary image processors and Kalman filtering. This research would later lead to the first European project on autonomous vehicles: Prometheus. Nowadays many ADAS have already been commercialized and can be found in the market. For example, the first adaptive cruise control (ACC) systems were introduced in high-end Lexus, Mercedes and Jaguar in the late 1990s [80]. ACC keeps a constant distance to the front vehicle by slowing or accelerating the host one. Lane Departure warning systems warn the driver when the vehicle moves out of its lane, unless the corresponding direction turn signal is on. This technology was first included in trucks in 2000 [81] and later in sedans. This technology is currently being improved by assisting the steering action or warning/intervening in lane changing in case of danger. One of the currently hot research topics are advanced front lighting (AFL) systems, which control the headlight parameters so that the beam is optimized for different conditions like driving speed and direction.

The reader is referred to books [23, 160] for more details on these systems.

## 1.2 Pedestrian protection systems

As can be seen, the improvements in automobiles safety have been typically focused on vehicle-to-vehicle crashes and protection of vehicle passengers. On the contrary, road users like pedestrians or bicyclists have not received as much attention as car occupants. Nevertheless, by having a look at the statistics it can be seen that the proportion of pedestrians killed in accidents is considerably high (Fig. 1.2). **In 2003 there were reported almost 150 000 injured and 7 000 killed pedestrians in European Union roads [54], representing the second source of injuries and fatalities, just after four-wheeled vehicle passengers.** The numbers for the United States are similar, counting 70 000 injured and 4 000 killed [55]. In low and middle income countries this number can neither be neglected. As an example, just Delhi City (India) registered almost 1 000 killed pedestrians in 1994 [113], and at the moment this number is likely to have grown considerably given that the number vehicles has been doubled in the country [120].

In view of of these terrible statistics, during the last twenty years companies have

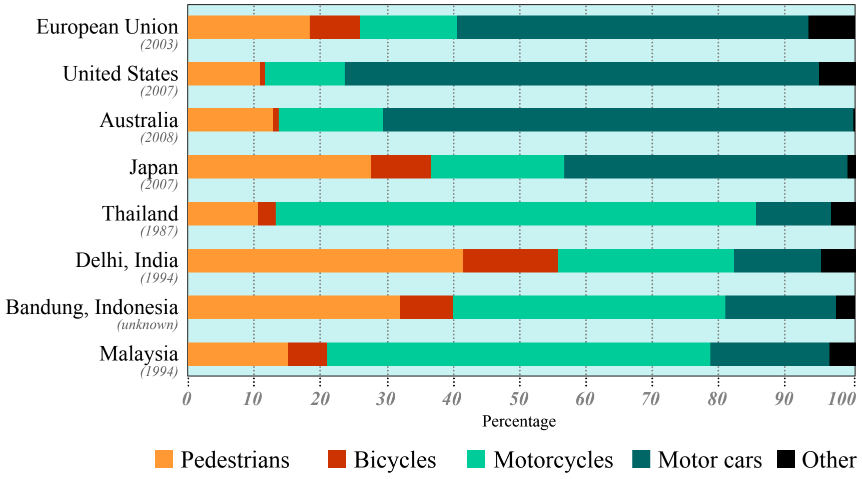


**Figure 1.2:** People killed in road accidents in different modes of transport as a proportion of total number of fatalities. Motor cars refers to occupants of motor vehicles with four or more wheels (from cars, trucks, buses, etc.). Motorcycles refers to two or three-wheeled motor vehicles occupants, including mopeds. Source of data: EU [54], USA [55], Australia [7], Japan [151], Thailand, India, Indonesia and Malaysia from [113]. Bar plot inspired in [127].

progressively turned their safety efforts also to pedestrian protection. At the beginning, research was focused on optimizing the physical parts of the vehicle in order to minimize impact severity. Some examples of this research direction, often referred to as improving safety through design, are collapsing fenders, hood and windshield, or increasing the space between (a softer) hood and the engine to accommodate the pedestrian's head in the case of a crash. The first investigations on intelligent systems addressing pedestrian protection were conducted in the 1990s by Papageorgiou (MIT), Gavrila (University of Amsterdam and Daimler Chrysler), Broggi and Bertozzi (University of Parma). Nowadays, pedestrian safety has become an interesting research and development topic for companies, governments and research centers. Some examples of such interest can be seen in the last three European Union Programmes for research, usually referred as Framework Programmes (FP), detailed next.

- Under 5th FP: PROTECTOR (4.4 million €, 2002-2003) and SAVE-U [136] (8 million €, until 2005), with Faurecia as coordinator, and CEA, Volkswagen AG, Daimler Chrysler AG, Siemens VDO Automotive AG and Mira Ltd as partners.

- Under 6th FP: PReVENT's APALACI [131] (3.75 million €, until 2008), coordinated by FIAT with partnering from Daimler Chrysler AG, Robert Bosch GmbH, Ibeo Automobile Sensor GmbH, Volvo and University of Parma.

- Under 7th FP: ADOSE (10.2 million €, 2008-2010), coordinated by FIAT; and FNIR [53] (3.12 million €, 2008-2010) coordinated by Autoliv AB.

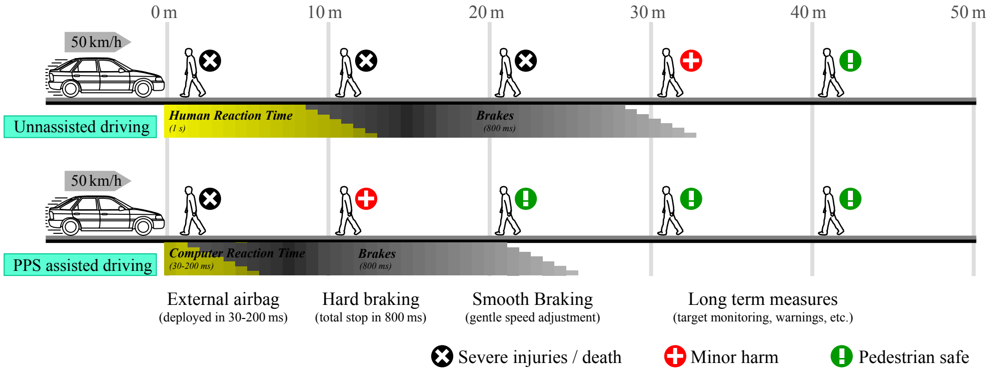


**Figure 1.3:** Comparison between unassisted and assisted driving. In the second case, typical safety measures are specified according to the distance of the pedestrian to the vehicle. Braking distance has been computed with the equation $\frac{v^2}{2G(f \pm s)}$, where $v$ is the initial speed (13.8 $m/s$), G is the gravity acceleration (9.8 $m/s^2$), $f$ is the friction coefficient (in this case 0.7, but can range from 0.5 to 1 depending on the weather, tires and asphalt) and $s$ the road slope (in this case assumed 0).

Pedestrian Protection Systems (PPSs) are a particular type of ADAS devoted to pedestrian safety. **A PPS is formally defined as a system that detects both static and moving people in the surroundings of the vehicle (typically in the front area) in order to provide information to the driver and perform evasive or braking actions on the host vehicle if needed.** Pedestrian detection before the impact (either long or short term) is crucial given that the severity of injuries for the pedestrian decreases with speed of the crashing vehicle. Thus, any reduction in the speed can drastically reduce the severity of the crash. According to [6], pedestrians have a 90% chance of surviving to car crashes at 30km/h or below, but less than 50% chance of surviving to impacts at 45 km/h or above.

Figure 1.3 illustrates the potential of PPSs. It shows a typical scenario with a vehicle moving at 50km/h and the possible harms to pedestrians at different distances. Without assistance (top), the human reaction time is long and consequently the brakes are actioned about 1 second after the dangerous situation[1]. A pedestrian is likely to suffer severe harm if he or she is at less than 25 m. With assistance (bottom), the benefits are twofold. First, they can reduce the reaction time to 100 ms or less [66, 141]. Second, since they can anticipate the potential accident they can not only provide warnings to the driver in a reduced time but also control the different active measures like airbags or brakes. Hence, the distance where pedestrians can be severely damaged is significantly reduced.

[1] In this case we have assumed a delay of 1s between the stimulus (e.g., a pedestrian crossing) and the brake actioning. However, reaction time depends on many variables: driver related (age, hours of continuous driving, consume of alcohol) and scene related (how discriminable the stimulus is, presence of distractors, day/night) [153].

## 1.3 The role of Computer Vision

The central problem of PPSs corresponds to the task of detecting pedestrians. In order to detect objects (e.g., vehicles, pedestrians, obstacles) in the distance, ADAS make use of sensors that provide data to a computer/controller that processes them and performs the corresponding actions. A comprehensive analysis of these sensors is made in Chap. 2. As will be seen, the most widely used sensors for pedestrian detection are cameras working either in the visible or infrared spectra, mainly thanks to the rich information they provide, with cues like edges, contours, texture or even relative temperature in the case of infrared cameras. Therefore it is clear that Computer Vision plays a key role in the task of pedestrian detection, which in fact is the central problem in PPSs.

Once the problem and sensors involved in detection have been introduced, the challenges for PPSs can be summarized in the following points:

- **Appearance variability** is very high in pedestrians, given that they can change pose, wear different clothes, carry different objects and their range of sizes (especially height) is considerable. See Fig. 1.3.
- Pedestrians shall be identified in **outdoor urban scenarios**, that is, they shall be detected in a cluttered background (urban areas are more complex than highways) under different illumination and weather conditions that add variability to the quality of the sensed information (e.g., shadows and poor contrast in the visible spectrum). In addition, pedestrians can be partially occluded by different urban elements such as parked vehicles or street furniture.
- Pedestrians shall be identified in very **dynamic scenes** given that not only the pedestrians move but also the camera does, which makes tracking and movement analysis difficult. Furthermore, pedestrians appear under different viewing angles (e.g., lateral and front/rear positions) and a big range of distances shall be reached. Figure 1.3 illustrates the risk areas in PPSs. Most of the systems are focused on the distances from 5 to 25 m to the camera, namely *high risk* area (a pedestrian at 25 m corresponds to a 30 × 60 pixels pedestrian with a 6mm focal length 640 × 480 pixels camera). However, extending the detection to 50 m, that is, covering also the *low risk* area, represents a great aid for PPSs in the long term accident prevention, as can be seen in Fig. 1.3.

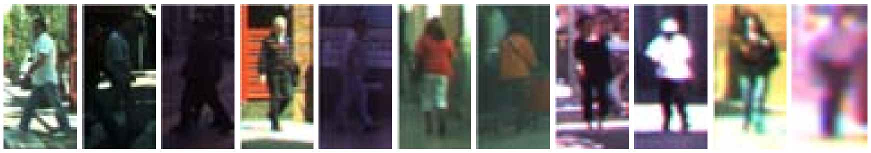


**Figure 1.4:** The variability of pedestrians is high as a result of the different possible illuminations, sizes, poses, view angles, clothes, etc.

- Nighttime detection with infrared cameras is affected by temperature, distance, as will be explained in Chap. 2.

- The required **performance** is quite demanding in terms of system reaction time and robustness (i.e., false alarms vs misdetections).

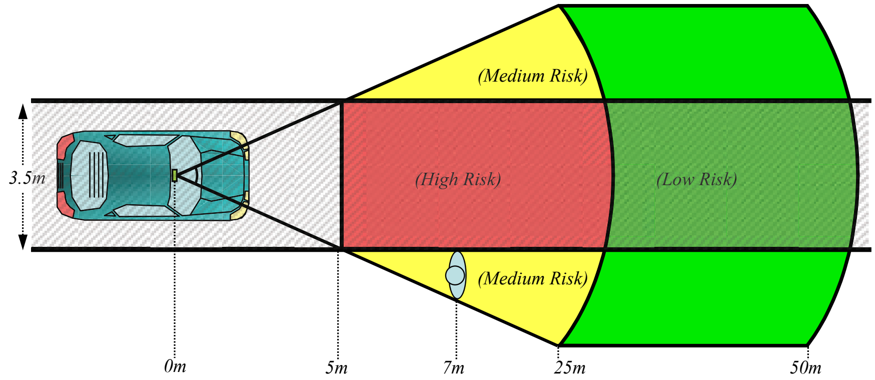


**Figure 1.5:** Typical risk areas in PPSs.

As can be appreciated, the topic differs from general human detection systems like surveillance or human-machine interfaces. Although PPSs can indeed make use of the techniques developed for these applications, many typically used simplifications must be discarded attending to the inherent challenges of PPSs:

- **Static camera assumption**, common in surveillance, is not applicable. Hence, techniques related to background subtraction are not useful in this research.

- **In-door illumination**, common in human-machine interfaces, does not fit driving assistance applications.

- **Model size constrains**, which are typically more constrained in dataset retrieval, are harder in PPSs. For example, the human model in visual dataset searching is normally focused on well-seen people with a considerable amount of pixels to analyze. In the case of PPSs, pedestrians at 50m can measure up to 10 pixels depending on the focal length. In the case of retrieval, however, the pose variability is more flexible than in PPSs, in which pedestrians are assumed to stand up on the road or pavement.

We refer the reader to the surveys in [111] and [56] for more details about human detection for applications different than PPSs.

## 1.4 Generic framework

The first proposals in pedestrian detection were primarily focused on the problem of classification by borrowing the sliding-window approach of other object detection areas (e.g., faces). However, they soon started to include other stages aimed at both reducing the number of false positives and to accelerate the processing. For example, tracking techniques are also being included to the systems recently.

We propose a generic architecture to be used as a framework first to review the literature in an ordered manner and second to organize the studied techniques along the thesis. In this way we will be able to analyze the techniques from a global viewpoint but following a well-defined criterion. The architecture consists of six conceptual modules each one with its own responsibilities:

- **Preprocessing**, which takes the input data from the camera and prepares it to the further processing, such as exposure time, gain adjustments and calibration, to mention a few.

- **Foreground segmentation** extracts regions of interest or candidates from the image to be sent to the classification module, avoiding as many background regions as possible.

- **Object classification**, which receives a list of candidates likely to contain a pedestrian. In this stage, they are classified as pedestrian or non-pedestrian with the aim of minimizing the number of false positives as well as the false negatives.

- **Verification and refinement.** Many systems contain one step that verifies and refines the ROIs classified as pedestrians, referred to as detections. The verification filters false positives using criteria not overlapped with the classifier while the refinement performs a fine segmentation of the pedestrian (not necessarily silhouette-oriented) so to provide an accurate distance estimation or to support the following module, tracking.

- **Tracking**, which follows the detected pedestrians along time with several purposes such as avoiding spurious false detections, predict the next pedestrian position and direction and even other high-level tasks, like inferring pedestrian behavior.

- **Application**, which takes high level decisions by making use of the information provided by the previous modules. This module represents a complete area of research, which includes not only driver monitoring or vehicle speed but also psychological issues, human-machine-interaction, etc.

Figure 1.6 shows a schematic overview.

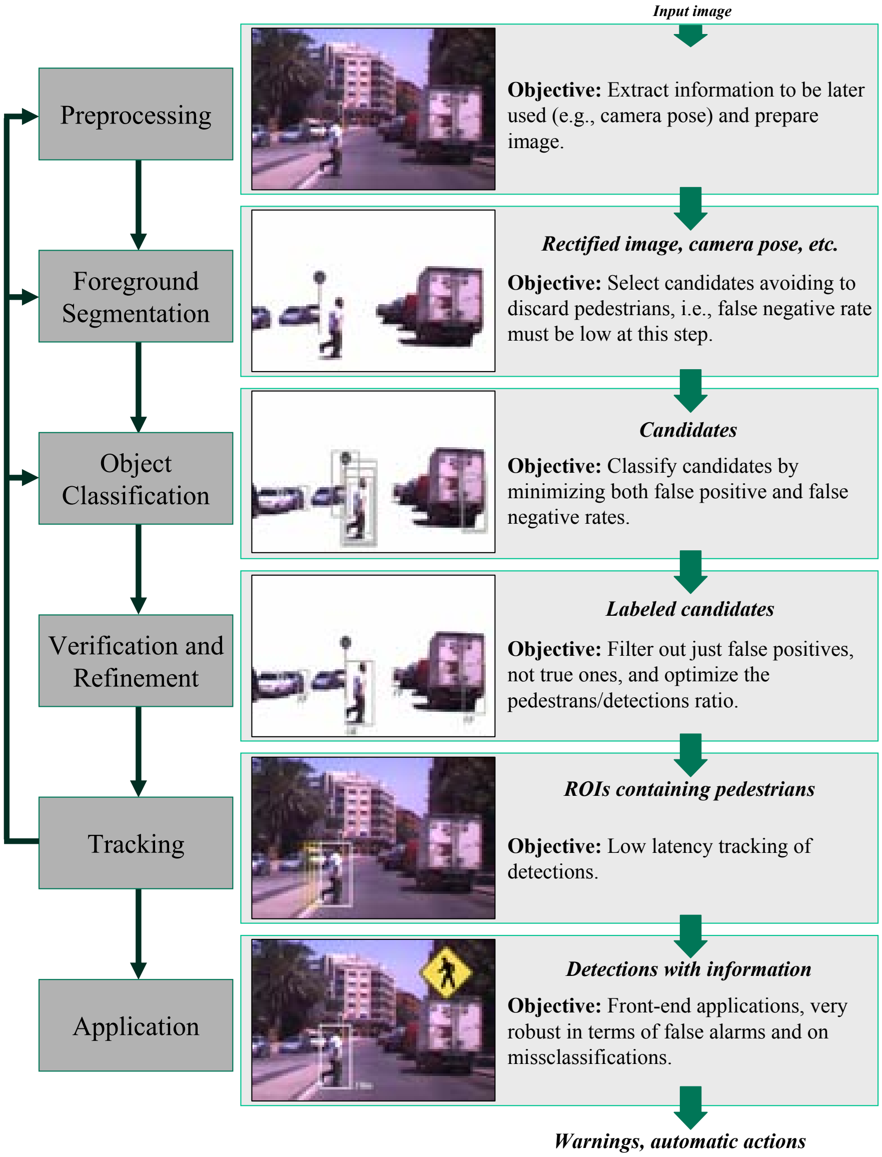

Input image
Preprocessing
Objective: Extract information to be later used (e.g., camera pose) and prepare image.
Foreground Segmentation
Rectified image, camera pose, etc.
Objective: Select candidates avoiding to discard pedestrians, i.e., false negative rate must be low at this step.
Object Classification
Candidates
Objective: Classify candidates by minimizing both false positive and false negative rates.
Verification and Refinement
Labeled candidates
Objective: Filter out just false positives, not true ones, and optimize the pedestrans/detections ratio.
Tracking
ROIs containing pedestrians
Objective: Low latency tracking of detections.
Application
Detections with information
Objective: Front-end applications, very robust in terms of false alarms and on missclassifications.
Warnings, automatic actions

**Figure 1.6:** The architecture proposed for an on-board pedestrian detection system, exemplified for the case of using a camera sensor working in the visible spectrum. The diagram is a simplification that covers most of the systems structure, so particular module organizations presented in some papers, for example interchanging tracking and verification stages, have not been included. However, potential feedback between modules (e.g., tracking-foreground segmentation) is getting more common so it has been illustrated by arrows.

## 1.5 Objectives

In this thesis we approach the problem of pedestrian detection from a global viewpoint. Instead of focusing on improving specific tasks as it is frequent in the literature, our research approaches the study from a more global perspective. The contributions of the thesis are summarized next:

- **A generic architecture** is proposed in this chapter with the aim of providing a conceptual division of the different tasks present in any PPS.
- A comprehensive **review of the literature** that makes use of the aforementioned architecture studies and analyses the large amount of existing techniques. The survey represents a crucial part of the research in the sense that it helps to visualize what has and has not been made and the current needs in this area.
- A study of **foreground segmentation algorithms** is carried out. Since this is a novel area, a novel dataset and protocol are first introduced. Then, we formalize the existing algorithms, propose new ones and evaluate them using the proposed dataset.
- **Classification aspects**. Instead of focusing the research on just outperforming classifiers as stand-alone entities with new features or classification algorithms, we explore general issues such as their generic requirements to be used along with the proposed foreground segmentation algorithms and their performance in the context of the system.
- An **analysis of window clustering** with respect to the foreground segmentation and object classification outputs is made based on two clustering algorithms.
- A **silhouette extraction technique** that does not require explicit training shape annotation is presented. As will be seen later, such a technique has not been much explored, and the unsupervised aspect has promising advantages.
- **Full system on complex scenes**. We present a system that combines some of the proposed techniques along the thesis, which is evaluated on complex urban traffic sequences. It is not a complete detection system since it does not exploit temporal coherence (i.e., tracking is out of the scope of the thesis) but we aim to give a feeling of the current possibilities of PPSs.
- We make public **new datasets** in complex realistic scenarios specialised in different modules: foreground segmentation, classification and system.

Of course, there are many interesting aspects that are left unexplored since they are out of the thesis scope. We think that enumerating some of these aspects can help to provide a better focus on the aim of the thesis. They are summarized next:

- Although a very strong emphasis in time consumption and realistic computational requirements is made along the thesis, and in fact is a key piece of it, we do not spend efforts on realtime optimizations.

- Nighttime pedestrian detection is of course included in the survey chapter. However, we do not explicitly test the proposed algorithms on night imagery. Of course, it is clear that in some cases they are still applicable (e.g., classification using infrared images or refinement techniques), but this is left for future research.

- As has been said before, we will not concentrate on improving the state of the art in classification, specially in improving generic high resolution human classification techniques (e.g., papers outperforming the INRIA benchmark [35]). In fact, any advance in classification algorithms is useful and in most of the cases it will fit the proposals and conclusions of this thesis.

- The algorithms are not specifically trained with children examples. As will be seen, the foreground segmentation is thought to work on adults, although the system will be able to detect pedestrians of very different sizes and proportions. Hence, children younger than 10 years old are not taken into account in the statistics, not for good nor for bad. Young children are expected to go with an adult, as in fact it is seen in the presented sequences, and according to NHTSA [55], children younger than 10 represent just the 4% of the killed and the 10% of the injured, so they are also left for future investigation.

## 1.6 Thesis outline

The thesis is organized in the following chapters. Chapter 2 presents the state of the art review. Chapter 3 focuses on the study of foreground segmentation techniques study. Chapter 4 studies different classifiers on a global viewpoint, specially analyzing their requirements and their performance when combined with foreground segmentation algorithms. In Chap. 5 a study of two different detection clustering techniques is made, together with the proposal of a novel silhouette extraction technique to be used as a refinement algorithm. Chapter 6 presents the results of proposed pedestrian detection system that makes use of several techniques presented in the thesis. Chapter 7 provide formal conclusions, formal discussion and perspectives on the research area.

In addition to the chapters, we attach several appendices that, far from being secondary or less important than the main chapters, include aspects that would interrupt the flow of the thesis as it is written. Appendix A describes the acquisition system used to record the sequences from which the evaluation datasets are extracted. Appendix B details the proposed pedestrian detection datasets. Appendix C presents a set of tables summarizing the most relevant PPSs with the aim of not only condensing the information of Chap. 2 but also providing additional information to it. Finally, App. D summarizes the existing performance evaluation measurements, plots, and protocols, and point the ones used along this thesis.

*The Dance of the Pan-Pan at the Monico*
Gino Severini, oil on canvas, 1909-1911
(Private Collection)

Gino Severini is one of the pioneers of Futurism, an art movement that highlights dynamism, speed and technology in opposition to previous more figurative art like Romanticism. In the painting, which is built on contrasted colors and straight traces, the reader can appreciate the movement and the surrounding effect that gives the idea of painting non-separate objects. For us, such a dynamic and almost chaotic scene greatly resembles to the existing literature on pedestrian detection we found at the beginning of the thesis in 2004, in which it is difficult to distinguish the foreground from the background at first glance, in which ideas are mixed together and no evident separation exists between them. Up to some extent, the idea behind a review is just the opposite to what Futurists pursued, that is, instead of totally braking with the figurative past and start a new movement based on the aforementioned ideas in a nearly abstract scenario, a review is entirely based on analyzing a chaotic past to construct a structured future.

# Chapter 2

# State of the art

This chapter presents a comprehensive review of the state of the art in pedestrian detection for PPSs. Given that the amount of papers in the literature is big, a typical survey enumerating the techniques one after another would result in a long and useless revision. Therefore, in this review we follow a different methodology: we split the different techniques according to the generic architecture introduced in Chap. 1. This procedure has two main benefits compared to the one-by-one survey. First, given that usually there are papers that make use of similar techniques, they are likely to be grouped and explained together, resulting in a shorter and more understandable analysis. Second, since the review is focused on the techniques exploited in each module it is easier to see the advantages/disadvantages in specific problems (e.g., candidates generation, tracking, etc.), which leads to a more fruitful analysis than if the problem is taken as a whole. Although not all the proposed modules are present in all the surveyed works and some of them can appear grouped by just one algorithm, we think that most of the systems can be conceptually broken down and fitted in this architecture so to make an easier comparison. In fact, such a break down is necessary for any complex system so to provide a deeper analysis. For instance, in the vehicle detection review by Sun et al. [147], the techniques are divided into hypothesis generation and hypothesis validation, thus allowing the reader to concentrate on the methods to solve simpler problems, not taking the problem as a whole.

The survey starts by an overview of the different sensors available in ADAS in Sect. 2.1, describing the properties of the ones used for PPSs in detail. Then, the review is made following the modules architecture in Sect. 2.2. In this case, each module is divided in review and analysis, the first enumerating the existing techniques and the later highlighting the advantages, disadvantages and future trends for each stage. Note that not all the reviewed techniques are strictly used in PPSs but we also include the ones we find of special importance for the area. For the sake of completeness, we also include a subsection (2.1) to review some systems exploiting the so-called sensor fusion. Benchmarking aspects are described in Sect. 2.3. Finally, a general discussion is presented in Sect. 2.4. In addition, with the aim of taking details out of the main text and presenting them in a visual comparative manner we include a set of tables summarizing the most relevant PPSs and interesting proposals in App. C.

## 2.1 Sensors

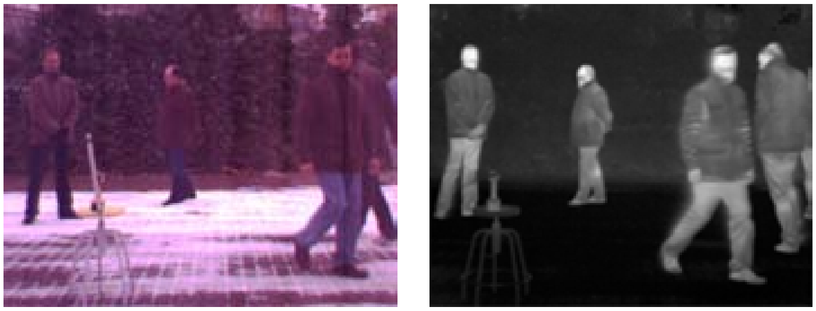
ADAS can be based on active or passive sensors. Active sensors transmit signals and observe their reflection from the objects present in a working space confined in an horizontal plane, that is, they typically work line-scanning. For instance, for ACC we can find radar (RAdio Detection And Ranging) approaches in which the signal are radio waves reaching detection distances of about 150 m with horizontal field of view (HFOV) of 16°. ACC also make use of laser-based approaches (Light Amplification by the Stimulated Emission of Radiation) in which the signal is infrared light, with examples like lidar (LIght Detection and Ranging), with detection distances of 100 m for 18° HFOV, and the so-called laserscanners, which reach 250 m with HFOV of 270°. In some cases these sensors are engineered to provide not only 1D but also 2D maps by attaching several line-scan planes. Passive sensors capture light by making use of matricial-scan chips, such as cameras (CCD or CMOS), and can operate either in the visible or infrared spectra. In [100] an ACC system based on a camera operating in the visible spectrum.

Each type of sensor has its advantages and disadvantages. For instance, active sensors are very good at estimating distances in real-time, while cameras operating in the visible spectrum not only have the advantage of a high spatial resolution (both horizontal and vertical) but also provide interesting information from cues like texture or color. However, cameras, as well as lidar and laser scanners, suffer in adverse conditions like in the presence of fog or heavy rain, while radar offers robust performance in virtually all weather conditions.

In the case of pedestrian detection, passive sensors are by far the most used. First, because it is hard for active sensors to distinguish pedestrians from other objects in urban environments given the reflectance properties of the former. Second, from a business point of view, active sensors tend to be more expensive than passive sensors. For instance, an ACC based on lidar is sold for about €500, one based on radar for about €1 500, and the price of a laser scanner is about €15 000, while a camera-based ACC can be probably sold for about €60.

Cameras can be divided according to their working range in the electromagnetic spectrum. Visible spectrum (VS) is in the 0.4-0.74 $\mu m$ range, near infrared (NIR) covers 0.75-1.4 $\mu m$ and thermal infrared[1] (TIR) captures in 6-15 $\mu m$. Figure 2.1 illustrates the same image in VS and TIR for an in-door scenario. However, as will be seen in this chapter, these ideal laboratory conditions differ from the real outdoor ones used in actual PPSs. Pedestrian detection is typically focused on daytime, hence VS cameras are the most extensively used ones. Some papers make also use of NIR, as will be seen later, and in fact they are cheaper than TIR ones (most VS cameras also capture NIR if the correct headlights are used, e.g., modern xenon technology). TIR cameras capture relative temperature, which is very convenient for distinguishing hot targets like pedestrians or vehicles from cold ones like asphalt or trees, hence they are used for pedestrian detection at night. Without enough ambient light, VS cameras provide too dark and poorly contrasted scenes, so pedestrian detection is not possible. Along the review we will assume that VS sensors are used if not stated the contrary.

[1] Also known as night vision, long wave infrared, far infrared (FIR) or infrared alone.



VS TIR

**Figure 2.1:** Appearance of VS and TIR for the same scene. As can be seen, the former provides color and textures while the latter highlights the higher temperature of the human body (Photo by ADAS Group, Universitat Autònoma de Barcelona and Universidad de Alcalá de Henares).

## 2.2 Literature review

### 2.2.1 Preprocessing

The preprocessing module includes tasks such as exposure time, gain adjustments and camera calibration, to mention a few.

#### Review

Although low-level adjustments, such as exposure or dynamic range are normally not described in ADAS literature, some recently published papers have targeted image enhancements for these systems. Real-time adjustments are a recurring difficulty, specially in urban scenarios. For example, short tunnels, narrow streets and the fast motion of the scene (common conditions in PPSs) can result in images with over/under saturated areas or poorly adjusted dynamic range, which creates additional difficulties for the latter algorithms of the system. Although not specifically devoted to ADAS, Nayar et al. [116] present some approaches for performing a locally adaptive dynamic range: fusion of different exposures, spatial filter mosaicing and pixel exposures, multiple image/pixel sensors, etc. Besides, during last years solutions exploiting high dynamic range images [104, 85] are gaining interest in driver assistance due to their potential to provide high contrast in the aforementioned scenarios. In fact, these cameras cover both VS and NIR spectra so they are also useful for nighttime vision.

Camera calibration is another main topic in the processing module. Few approaches tackle both intrinsic and extrinsic on-board self-calibration [25, 37]. The most common approach is to initially compute the intrinsic parameters and then to assume that they are constant, while the extrinsic parameters are continuously updated. This procedure, which is often referred to as camera pose estimation, avoids the so-called constant road slope assumption, a simplification that is not applicable to real PPSs given the road slope variability in urban scenarios and the changes in vehicle dynamics.

The existing approaches can be divided into two categories: monocular-based and stereo-based. In the former case, the algorithms are mainly based on the study of visual features. In [12, 30], Broggi et al. correct the vertical image position by relying on the detection of horizontal edges oscillations: the horizon line is computed according to the previous frames. A comparative study of different monocular camera pose estimation approaches is presented in [24]. It includes horizontal edges, features-based and frame difference algorithms. Recently, [74] presents a probabilistic framework for 3D geometry estimation based on a monocular system. A training process, based on a set of 60 manually labeled images, is applied to form a prior estimation of the horizon position and camera height (i.e., camera pose values).

Regarding stereo-based pose estimation, Labayrade et al. introduce v-disparity space [89], which consists of accumulating stereo disparity along the image y-axis in order to 1) compute the slope of the road (which is related to the horizon line) and 2) point out the existence of vertical objects when the accumulated disparity of an image row is very different from its neighbors (Fig. 2.2(*d*)). Extensions of this representation can be found in [76]. Other approaches work in Euclidean space. For instance, Sappa et al. [134] propose to fit 3D road data points to a plane, whereas Nedevschi et al. [117] use a clothoid. In the Euclidean space classical least squares fitting approaches can be followed, while in the v-disparity space, voting schemes are generally preferred (e.g., Hough transform). Recently, Ess et al. [46, 45] have proposed the use pedestrian location hypotheses together with depth cues to estimate the ground plane, which is used to reinforce new detections-tracks. The authors call this approach cognitive feedback, in the sense that a loop is established between the classification and tracking modules with the ground plane estimation.

### Analysis

High dynamic range sensors provide the possibility of obtaining highly contrasted images in outdoor scenarios. In the next years, this technology will be of crucial importance in PPSs in order to avoid the over/under-saturated regions that are typically seen in ADAS imagery. In fact, many of the failures of the current detection algorithms is related to poorly contrasted images so this technology will undoubtedly benefit the system performance (see datasets in Sect. 2.3).

Stereo-based approaches provide more robust results in camera pose estimation than monocular approaches. Horizon-based stabilizers are based on the assumption that the changes in the scene are smooth, which is not always a valid assumption in urban scenarios. Moreover, in such monocular-based approaches, the global error increases with time as long as the estimation depends on previous frames (the drift problem). On the contrary, stereo-based approaches (both disparity and 3D data) do not accumulate errors and can provide information about the object's distance from the vehicle. It is not clear whether disparity-based approaches are better than 3D-based approaches. Each approach presents advantages, disadvantages and limitations. For example, disparity-based approaches are generally faster than those based on 3D data points are; however they are limited to planar road approximations while 3D-based approaches allow plane, clothoid and any free form surface approximation.

The more recent of the reviewed works show a clear trend towards using stereo-based approaches to obtain accurate camera pose estimates in spite of the additional CPU time required for disparity/depth estimation.

### 2.2.2 Foreground segmentation

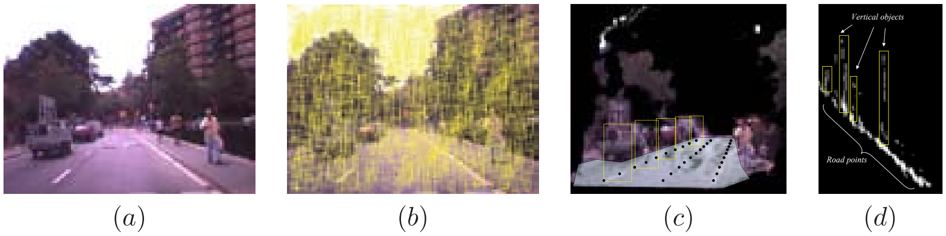
Foreground segmentation, sometimes referred to as *candidate generation*, extracts *regions of interest* (ROIs) from the image to be sent to the classification module, avoiding as many background regions as possible. In this thesis the term candidate generation is used to refer to the methods that extract specific windows in the image, while foreground segmentation is seen as the generic task of segmenting foreground from background, either extracting windows or just pointing to rough regions of interest. Although some papers do not contain a specific segmentation module (e.g., sliding window), these techniques are of remarkable importance not only to reduce the number of candidates but also to avoid scanning regions like the sky. The key to this stage is to avoid missing pedestrians; otherwise the subsequent modules will not be able to correct the error. While describing this module we will often use the term *pedestrian size constraints* (PSC), which refers to the aspect ratio, size and position that candidates shall fulfill to be considered to contain a pedestrian (e.g., in [68] pedestrians are assumed to be around 1.70 m high, with some standard deviation, 20 cm, tall with a 1/2 aspect ratio, hence ROIs are constrained to these parameters).

#### Review

The simplest candidate generation procedure is an exhaustive scanning approach [35, 124] that selects all of the possible candidates in an image according to PSC, without explicit segmentation. This method is known as sliding window. For instance, in [35], the authors start by scanning the image with candidate windows of 64 × 128 pixels, placing these windows every 8 pixels. Then they reduce the image size by a factor of 1.2, and perform the same scan again. This procedure has two main drawbacks: 1) the number of candidates is large (see Fig. 2.2(*b*)), which makes it difficult to fulfill real-time requirements, although some proposals have recently studied this problem [175, 163, 173]; and 2) many irrelevant regions are passed to the next module (e.g., sky regions or ROIs inconsistent with perspective), which increases the potential number of false positives. As a result, other approaches are used to perform explicit segmentation.

**2D-based:** Miau et al. [108, 79] use a biologically inspired attentional algorithm that selects ROIs according to color, intensity and gradient orientation of pixels. In several works from *Parma University*, the vertical symmetries in the visible [27, 26, 13, 16] and TIR spectra is used alone [15, 12] or as a complement to stereo imagery [26]. In this case, candidates are adjusted around each symmetry axis maintaining the PSC. The presence of many horizontal edges is often taken into account as a non-pedestrian quality.

Intensity thresholding is the most intuitive segmentation technique when working with TIR images. Some implementations include single thresholding [145]; double im-



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

**Figure 2.2:** Foreground Segmentation Schemes. (a) Original image. (b) Exhaustive scan [35] (just showing 10% of the ROIs). (c) Sketch of road scanning after road fitting in Euclidean space [68]. (d) Results of v-disparity applied to the same frame [89].

age and hot spots-based thresholding [29] and adaptive intensity-based thresholding [154, 149]. Another simple technique consists in the use of vertical and horizontal histogram projection together with thresholding [49, 19]. Hypermutation networks [119], which use a multi-stage neighborhood pixel classification, are considered in more sophisticated approaches like [107, 101] to classify pixels as foreground/background. In [107], the output pixels from the network are grouped by using connected component analysis so the algorithm can be understood as a segmentation/classification process.

**Stereo:** Franke et al. [60] present one of the first stereo algorithms specifically developed for ADAS. Local structure classification, which resolves ambiguities by the use of a disparity histogram, is used to perform stereo correspondence. They extend the algorithm with a multi-resolution method with sub-pixel accuracy in [57, 59]. These works become more relevant when they are used in the well-known PROTECTOR system [66]. In this system, the returned map is multiplexed into different discrete depth ranges, which are then scanned with PSC-windows, taking into account the location of the assumed ground plane. If the depth features in one of the windows exceed a given percentage, the window is added to the candidates list supplied to the classifier.

Many authors [27, 72, 87] have made use of the aforementioned v-disparity representation [89] to identify ground and vertical objects. Extraction of candidate regions bounding vertical objects is straight-forward after the removal of road surface points. These approaches are based on the fact that a plane (i.e., road surface) in Euclidean space becomes a straight line in the v-disparity space (Fig. 2.2(*d*)). In [68], we use stereo-based road plane fitting [134] to dynamically select a set of candidates that are lying on the ground and satisfying the PSC avoiding the flat road assumption (Fig. 2.2(*c*)). Disparity map analysis together with PSC is also used to extract candidates [174, 26, 144].

Recently, Krotosky and Trivedi propose the use of multimodal-stereo analysis to generate candidates, that is, combining two different sensor types, like VS and TIR to perform stereo [86] or a VS stereo pair matched with TIR imagery [87]. This approach, which corresponds to sensor fusion (detailed in Sect. 2.2.7), worths a mention here because of its potential to widen the range of working conditions, specially in the case of the tetra-camera configuration, consisting of a VS pair for daytime and a TIR pair for nighttime.

**Motion-based:** Inter-frame motion and optical flow [42] have also been used for foreground segmentation, primarily in the general context of moving obstacle detection. Franke et al. [58] proposed to merge stereo processing, which extract depth information without time correlation, and motion analysis, which is able to detect small gray value changes in order to allow early detection of moving objects. In [90], Leibe et al. present a real-time structure-from-motion based approach for ground plane estimation. This online estimated plane is used to update the camera calibration and thus to segment objects from the ground surface.

### Analysis

The exhaustive scan is typically used in general human detection systems, for example, image retrieval, whereas PPSs tend to use some kind of segmentation, as is shown in Table C.1. In fact, the latter can take advantage of some application prior knowledge (e.g., it is not necessary to search the top area of the image), so that the number of ROIs to process can be greatly reduced. For example, a typical exhaustive scan on a 640 × 480 image can provide from 400 000 to 3 000 000 ROIs, depending on the sampling step and the minimum candidate size. In contrast, as will be seen in Chap. 3, sampling just the estimated road can reduce this number to 20 000-40 000, again depending on the density of the scan. Furthermore, stereo-based segmentation can further reduce this number depending on the content of the scene.

According to the literature, stereo is the most successful option. 2D-based analysis does not provide convincing results at this stage. For instance, symmetry is not very reliable so extra-cues such as depth are necessary, hotspot analysis seems to be ruled by heuristics and attentional bottom-up pixel-based algorithms like [108] do not provide accurate ROI positions, so the reduction of the number of candidates is not as large as expected. More sophisticated appearance based techniques are likely to be used during classification, not during candidates generation. In addition, the accuracy of motion-based approaches depends on driving speeds, and the reliability of those approaches has not been demonstrated under the wide range of ADAS conditions.

Stereo-based systems present several advantages: 1) they have good accuracy in the working range of pedestrian detection; 2) they are robust to circumstantial variability (e.g., illumination in VS or temperature in TIR); and 3) they provide useful information for other modules (e.g., distance estimations for tracking) and other ADAS applications (e.g., free-space analysis [59, 10]). The drawbacks of such systems are as follows: 1) blind areas in non-textured regions; 2) slow speed (although advances in parallel data processing are being studied [156]); and 3) a requirement for postprocessing in order to separate regions with similar disparity and position to fit the pedestrian size and aspect ratio [174].

In conclusion, stereo is the primary option for future systems. Along with the aforementioned properties, stereo-pairs are improving in accuracy, computation time and resolution, facilitating the development of new systems.

It is clear that in such an oriented application problem the use of scene prior knowledge plays a key role. A few recent studies on more more sophisticated algorithms based on preattentive cues and context should be highlighted: Torralba et al. [73, 150] and Hoiem et al. [74, 75] groups. In these papers the important roles of perspective, scene object dependencies, surfaces and occlusions in object detection are demonstrated. In addition, active sensors (e.g., laserscanners), which can estimate distances without high computation times (Sect. 2.2.7) are also likely to be exploited in specific PPSs tasks (e.g., short-time collision detection).

### 2.2.3 Object classification

The object classification module receives a list of ROIs that are likely to contain a pedestrian. In this stage, they are classified as pedestrian or non-pedestrian with the goal of minimizing the number of false positives and false negatives.

#### Review

The approaches to object classification are purely 2D, and can be broadly divided into silhouette matching and appearance.

**Silhouette matching:** The simplest approach is the binary shape model, presented by Broggi et al. [26], in which an upper body shape is matched to an edge modulus image by simple correlation after symmetry-based segmentation. A more sophisticated approach is the Chamfer System, a silhouette-matching algorithm proposed by Gavrila et al. in [66, 65, 67]. This system consists of a hierarchical template-based classifier (Fig. 2.3) that matches distance-transformed ROIs with template shapes in a coarse-to-fine manner. The shape hierarchy is generated offline by a clustering algorithm. This technique has also been exploited for TIR images in [101]. Also in the TIR spectrum, Nanda et al. [115] perform probabilistic template matching on a multiscale basis, by using just three templates (each for a defined scale). In [29], Broggi et al. present two methods that rely on templates, one of which is based on simple matching and another based on legs position.

**Appearance:** The methods included in this group define a space of image features (also known as descriptors), and a classifier is trained by using images known to contain examples (pedestrians) and counter-examples (non-pedestrians). Table 2.1 summarizes the typical learning algorithms (classifiers) used in the literature.

Following a holistic approach (i.e., target is detected as a whole), in [66, 67] Gavrila et al. propose a classifier that uses image grayscale pixels as features and a *neural network with local receptive fields* (NN-LRF [114]) as the learning machine that classifies the ROIs generated by the Chamfer System. In [174], Zhao et al. use image gradient magnitude and a *feed forward neural network*.

Papageorgiou et al. [124] introduce the so-called Haar features (HFs) as features to train a quadratic support vector machine (SVM) with front- and rear- viewed pedestrians. HFs compute the pixel difference between two rectangular areas in different configurations (Fig. 2.4$(a)(b)(c)$), which can be seen as a derivative at a large scale. Viola and Jones [158, 82] propose AdaBoost cascades (layers of threshold-rule weak

**Table 2.1:** Typical learning algorithms used in PPS.

| | | |
|---|---|---|
| SVM [157] | Definition | Finds a decision boundary by maximizing the margin between the different classes. |
| | Properties | - Decision boundary can be linear.<br>- Data can be of any type, i.e., scalar or vector features, intensity images, etc. |
| | Used features | Intensity image [149, 168], Haar features [124, 112, 2], HOG [35, 145, 172, 164], Edgelet [172], Shape Cont. [164]. |
| AdaBoost [61] ([137, 62], [77, 169]) | Definition | Constructs a *strong* classifier by attaching *weak* classifiers (often rule-of-thumb) in an iterative greedy manner. Each new classifier focuses on misclassified instances. |
| | Properties | - Speed optimized thanks to the use of cascades.<br>- Can be combined with any classifier to find weak rules (e.g., with SVM [175])<br>- Few parameters to tune. |
| | Used Features | Haar features [158, 68, 101], HOG [175, 164], EOH [68], Edgelet [172, 166], Shape Context. [164] |
| Neural Networks [21] | Definition | Different layers of neurons (many different configurations are possible) provide a non-linear decision. |
| | Properties | - Many configurations and parameters to choose.<br>- Raw data is often used, i.e., no explicit feature extraction process is needed. |
| | Used Features | Intensity image [148], Gradient mag. [72, 174], LRF [114]. |

classifiers) as a learning algorithm to exploit Haar-like features (the original HFs plus two similar features, Fig. 2.4($d$)($e$)) for surveillance-oriented pedestrian detection. In this case, HFs are also exploited to model motion information. These features have been quite successful for object recognition. Mählisch et al. [101] combined Haar-like features with the Chamfer System in a system based on TIR imagery. As will be detailed in Chap. 3, we make use of Real AdaBoost to select the best features among a set of Haar-like and *edge orientation histograms* (EOH; [95]) features to classify ROIs in the VS [68]. EOH first compute the gradient magnitude of the image and then distribute the pixels into $k$ different bins (in this case $k = 4$) according to their gradient orientation. The features are defined as the ratio between the summed gradient magnitudes of two bins for a given rectangular region. For example, the feature $\frac{\psi_{(0,\pi/4)}}{\psi_{(\pi,3\pi/4)}}$, where $\psi_{a,b}$ corresponds to the summed gradient magnitude of pixels laying in the specified $(a, b)$ angle interval, gives one real value, which is fed as a feature to the threshold rule classifier. Both Haar-like features and EOH can make use of the *integral image representation* [158], which computes the sum of pixels of a region in just four memory accesses.

Dalal and Triggs [35] present a human classification scheme that uses SIFT-inspired [99] features, called *histograms of oriented gradients* (HOG), and a linear SVM as a learning method. A HOG feature also divides the region into $k$ orientation bins (in this case, $k = 9$), but instead of computing the ratio between two bins, they define 4 different cells that divide the rectangular feature, as illustrated in Fig. 2.6. In addition, a Gaussian mask is applied to the magnitude values in order to give more weight the center pixels, and the pixels are interpolated with respect to pixel location within a block (both factors disallow the use of the integral image). The resulting feature is a 36-dimensional vector containing the summed magnitude of each pixel cells, divided into 9 bins. These features have been extensively exploited in the literature. In [145], they are used for ADAS-oriented pedestrian detection in TIR images, while [36] uses them with optical flow images. Other papers propose new learning approaches by making use of the same features. Zhu et al. [175] use HOG as a weak rule of AdaBoost, achieving the same detection performance, but with less computation time; whereas Pang et al. [123] use Multiple Instance Learning (concretely, *Logistic Multiple Instance Boost* [169]) together with weak classifiers based on graph embedding to model variations in pedestrians' poses and viewpoints. Recently, Maji et al. [102] have outperformed the state-of-the-art detectors by using multi-level edge energy features (similar to HOG, but simpler) and the Intersection Kernel SVM. First, they apply non-maximum suppression to each gradient orientation bin, which is then used to construct a pyramid of histogram features at different scales ($64 \times 64$, $32 \times 32$, $16 \times 16$, $8 \times 8$). Intersection kernels are then used to train these features on an SVM.

Wu et al. [167] study the performance of short segments (up to 12 pixels long) of lines or curves, referred to as *edgelets*, as features for AdaBoost for VS images. In this case, a mask is attached to each feature in order to provide pixel-wise segmentation (Fig. 2.5). The same authors study edgelets and HOG together with AdaBoost and SVM learning algorithms in both the VS and TIR [172]. Also exploiting local gradient orientation features, Sabzmeydani et al. [133] propose to use AdaBoost to model each $n \times n$ cell (they test different cell sizes, $n = 5, 10, 15$), with respect its orientation. Each of the selected cells is referred to as a *shapelet* feature.

Tuzel et al. [155] propose a novel algorithm that is based on the covariance of different measures (position, first and second-order derivatives, gradient module, gradient orientation) in subwindows as features, along with LogitBoost [62] using Riemannian manifolds. The achieved performance is comparable to state-of-the-art detectors, while the computation time is comparable to [35].

Other features and learning algorithms used in the literature include the gradient magnitude and quadratic SVM [72], Four Directional Features and Gaussian kernel SVM [144], and intensity image with Convolutional Neural Networks [148] or with an SVM [149, 168].

Part-based approaches, contrary to the previous techniques, combine the classification of different parts of the pedestrian body (e.g., head and legs), instead of classifying the entire candidate as a single entity.

Mohan et al. [112] use HFs and a quadratic SVM to independently classify four human parts (head, legs, right and left arms). The classifications of these parts are combined with a linear SVM. In [141], Shashua et al. use thirteen overlapping parts (Fig. 2.7), described by SIFT-inspired [99] features, and ridge regression to train the classifier of each part. The training set is divided into 9 clusters according to pose and illumination conditions, resulting in $9 \times 13 = 117$ classifiers, in order to deal with the high intra-class variability. The outputs of the classifiers are fed as weak rules to an AdaBoost classifier that sets the final classification rule. Wu and Nevatia [166] propose the use of four body parts (full-body, head-shoulder, torso and legs) and three view categories (front/rear, left profile and right profile) to train a nested-weak-classifier AdaBoost [77]. They use edgelets as features. In this case, Bayesian reasoning, together with a typical surveillance assumption (camera looking down the plane), is used to combine the body parts. In the case of [125], Parra et al. define the features as the co-occurrence matrix between Canny edges and normalized grayscale image, the orientation histogram, the magnitude and orientation of the image gradient, and the texture unit number, which are then fed to a SVM. Tran et al. [152] propose estimation of the the pedestrian pose in the ROI by the use of structure learning, which provides a tree parts configuration. After the estimation, the ROI conditioned by this configuration can be classified.

Felzenszwalb et al. [51] sum the classification score of the candidate and six different dynamic parts (which are not constrained to a unique position relative to the candidate). In this case, the authors use what they call *latent SVM* and HOG. Dollár et al. [40] extend the aforementioned Multiple Instance Learning to a part-based scheme called Multiple Component Learning, using Haar features, gradient magnitude and orientation features are used. Both approaches [51, 40] notably avoid the task of manually annotating parts since they are automatically determined by the method.

Lin and Davis [97] have recently proposed a technique that combines some of the aforementioned paradigms to a greater or lesser extent, i.e., silhouette, appearance, holistic and parts-based. First, HOG descriptors are computed for the whole image following [35]. Then, the descriptors are used to extract a silhouette, which is fed to a probabilistic hierarchical part-matching algorithm. Finally, HOGs are again computed for the closest regions of the matched silhouette, serving as features for a radial basis function (RBF) kernel SVM.

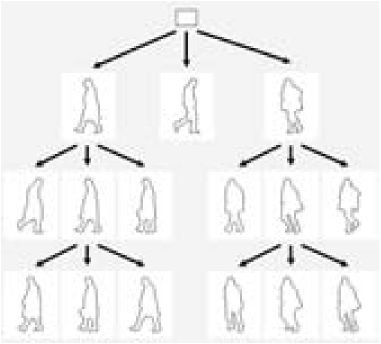


**Figure 2.3:** Hierarchy of templates used in the Chamfer System (figure from [65]).

**Other approaches:** Following recent research in object detection, Leibe et al. [92] present a technique termed the *implicit shape model*, which avoids the candidate generation step. The idea is to use a keypoints detector, Hessian-Laplace [109] in this case, then compute a shape context descriptor [11] for each keypoint and finally cluster them to construct a codebook. During recognition, each detected keypoint is

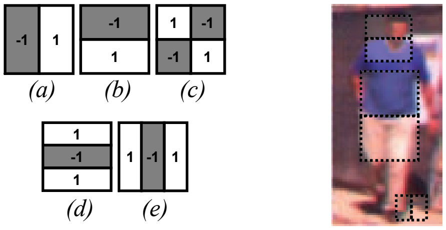


**Figure 2.4:** $(a - c)$ Haar features and $(a - e)$ Haar-like features, applied at specific positions of a pedestrian sample.

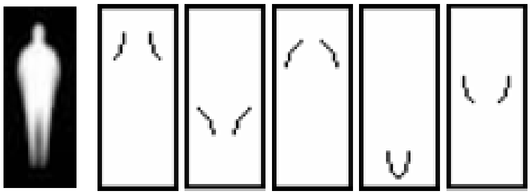


**Figure 2.5:** First five edgelet features selected by AdaBoost in the approach by Wu and Nevatia (figure from [167]).

matched to a cluster, which then votes for an object hypothesis using Hough voting, thus avoiding a candidate generation step. The Chamfer distance is used to provide a fine silhouette segmentation of the pedestrian. In [138, 139], Seeman et al. improve this technique with multi-aspect (viewpoint and articulation) detection capabilities, extending the hypothesis voting to object shapes, rather than just objects.

### Analysis

Silhouette matching methods are not applicable as stand-alone techniques. Even the elaborate Chamfer System needs an extra appearance-based step. In contrast, methods that exploit appearance seem to indicate the current direction of research, specifically revolving around the continuous development of new learning algorithms and features for use in these algorithms, not only in pedestrian detection but also in general object classification.

Despite the large number of papers, approaches tend to be poorly compared to one another in PPSs research. Wojek et al. [163] try shed light on the comparison of classifiers with a study on some popular features and learning methods. Two conclusions are highlighted: HOGs and shape context features are the best option, independent of the learning algorithm, and feature combination significantly improves detector performance. In recent years, however, the lack of comparisons has been amended thanks to Dalal's proposal (both detector and dataset [35]), which has been established as a *de-facto* baseline. In fact, many of the techniques proposed within the last two years [155, 152, 133, 102, 51, 40, 97] use this benchmark, which makes it feasible to gain insights into the proposed module.

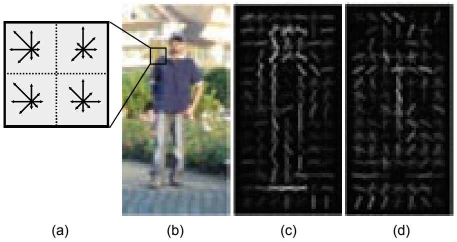


**Figure 2.6:** Histograms of Oriented Gradients by Dalal and Triggs (figure from [35]). (a) The descriptor block. (b) Block placed on a sample image. (c,d) HOG descriptor weighted by positive and negative SVM weights.

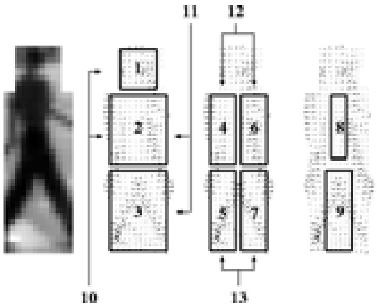


**Figure 2.7:** Part-based classification using gradient-based features by Shashua et al. (figure from [141]).

Given the number of papers presented during in recent years, it is not possible to point to one method as the best option. Nevertheless, some research directions are clearly gaining relevance. Holistic classifiers seem to have reached their performance limit, at least for current datasets, and are unable to deal with high variability. According to experiments, non-standard poses greatly affect their performance: beyond the usual straight versus crossing legs, pose variability also affected the head and torso alignment in the training examples. In addition, the diversity of poses causes many pedestrians to be poorly represented during training (e.g., running people, children, etc.). Parts-based algorithms that rely on dynamic part detection [51, 152, 139] handle pose changes better than holistic approaches. This information has been demonstrated to be beneficial in classification. Furthermore, other interesting ways to overcome this variability being explored (e.g., multiple instance learning), may provide additional benefits, like relaxing the annotation process. Of course, any improvement in existing algorithms, like the proposals in [102, 139], or new features that exploit typical measures (i.e., intensity, gradient, etc.) in new ways, like shape context [11] or HOG [35] will contribute to the improvement of these systems.

The real-time requirements of PPSs have been a principal restriction on the features and algorithms; however, there are proposals to reduce the high computation cost of some techniques. The disadvantage is that in most of the cases the opti-

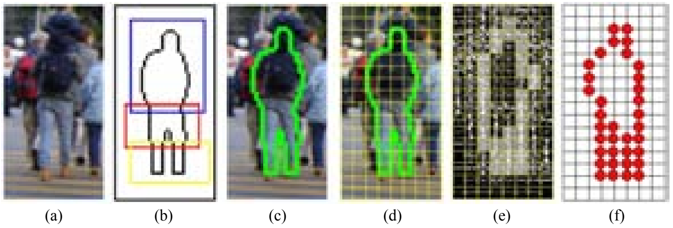


**Figure 2.8:** Pose invariant algorithm by Lin and Davis (figure from [97]). (a) Input image. (b) Part-template detections. (c) Pose and shape segmentation. (d) Cells grid used for HOG computation. (e) HOGs. (f) Cells relevant to HOG.

mizations are applicable only in specific cases under restricted cases of foreground segmentation and object classification algorithms with specific parameters.

Some interesting proposals can be found in [173], in which a multiresolution rejection scheme is used to boost the computation time to roughly 7 times faster than the original approach [35]; or [171, 163], in which authors present 10 and 34 times faster versions of [35] by the use of hardware-based implementations of the algorithm.

### 2.2.4 Verification and refinement

Many systems contain one step that verifies and refines the ROIs classified as pedestrians. The verification step filters false positives, using criteria that do not overlap with the classifier while the refinement step clusters the classifier output windows and segments (not necessarily silhouette-like) the pedestrian.

#### Review

Gavrila et al. [66, 67] verify detections by performing cross-correlation between the left image of a stereo pair and the isolated silhouette computed by the Chamfer system in the right image. In [57], Franke and Gavrila suggest the analysis of gait pattern of pedestrians crossing perpendicular to the camera. The target shall be tracked before applying this method, thus the order of verification/refinement and tracking modules is interchanged for this particular technique. Chamfer matching is used to both verify and refine the found pedestrian shapes in [94]. In [141], Shashua et al. propose a multi-frame approval process that consists of validating the pedestrian-classified ROIs by collecting information from several frames: gait pattern, inward motion, confidence of the single-frame classification, etc. In this case, verification follows tracking.

For refinement, one essential algorithm that provides one detection per target is clustering. Assuming that classifiers provide a peak at the correct position and scale of the target and weaker responses around it, Dalal [34] makes use of mean shift [33] to find the minimum set of detection windows that best adjust to the pedestrians in the image. For the sake of completeness, it is worth mentioning the work by Agarwal et al. [1]. Their proposal, which is tested for vehicle detection but likely to be applied in PPSs, consists of two algorithms. The first creates an activation map in which high-confidence detections mark their neighborhoods as invalid for new detections. Given that this system is based on a parts-based classifier, the second algorithm constrains the parts to be assigned to only one detection, and thus non-maximum detections are discarded by iteratively decreasing their confidence.

In [26], by Broggi et al., the silhouette of the head and shoulders that is matched during classification is taken as a reference for refining detection down to the feet by using the vertical edges computed for the symmetry detection. The accurate localization of the feet is then used to compute the distance to pedestrians by assuming a planar road. Then, stereo processing completes the refinement, by correlating the left-image window to certain positions of the right image.

There are just a couple of approaches to provide silhouette-based segmentation, which are extensions to methods described along this chapter. The first one is the Chamfer System [67], in which the matched shapes are directly used as final segmentations. The second is Leibe's *implicit shape model* [92], which is used as a segmentation method by adding an annotated binary mask to each training example. In this way, each detected keypoint has an associated mask, hence the final silhouette is reconstructed by grouping all the masks.

Some techniques that utilize TIR images are 2D model matching [12]; 3D [15, 28] model matching; symmetry [14]; and a multiple filter approach, based on the area overlap between positively classified candidates and group multiple detections in a single window [101].

### Analysis

This module should be a complement to the classification module. In fact, authors often refer to the described techniques as a two detection process, in the sense that verification algorithms are tied to the classification output, that is, to the characteristics of its false positives. For instance, a classifier that fails to discard trees will not gain much benefit from a verifier that just distinguishes vertical regions in 3D. It is important to note that stereo information tends to be used as long as the classification is based on a 2D image. In addition, it is reasonable to expect that as more cues are used in verification, the results will be richer, for example, stereo imagery may be combined with classification confidence, symmetry or gait. It should also be noted that the use of verification after tracking presents an interesting approach, since common movement-based techniques (e.g., gait pattern analysis) used in surveillance can potentially be applied.

The employed refinement methods should also be chosen according to the utilized foreground segmentation technique and the available information. Each of the methods present advantages and disadvantages. For instance, mean shift has proven to be a reliable technique for full-scan processing, but has not been evaluated for other foreground segmentation algorithms for which the ROI scan is not very dense (e.g., road sampling). The same applies to the nature of the classifier: it has not been proven how this clustering algorithm deals with imprecise classifiers.

Distance estimations from stereo images, when available, are a good cue for adjustment of the final detection size, but the error increases with the target distance. A study of the quality of the final detections in terms of road plane adjustment (i.e., pedestrian distance), bounding box accuracy, etc. under a set of different candidate generation schemes, refinement algorithms and cues (e.g., disparity, road plane adjustment, TIR symmetry) would be of great interest.

### 2.2.5 Tracking

The most evolved systems use a tracking module to follow detected pedestrians over time. This step has several purposes: avoiding false detections over time; predicting future pedestrians positions, thus feeding the foreground segmentation algorithm with candidates; and at a higher-level, making useful inferences about pedestrian behavior (e.g., walking direction).

#### Review

Franke et al. propose the use of two Kalman filters [59, 162], one controlling lateral motion (yaw rate of the own vehicle is used) and the other controlling longitudinal motion, to determine the speed and acceleration of detected objects. Later, in [66, 67] authors from the same research group used an $\alpha$-$\beta$ tracker (a simplified Kalman filter with pre-estimated steady-state gains and a constant velocity model) based on the window representation from their stereo verification phase. Three cues are used: the Euclidean distance between bounding box centroids, shape dissimilarities (to avoid multiple tracks for single objects) and the Chamfer distance (to avoid multiple objects assigned to single tracks). Also using Kalman filters as tracking filters, [16] uses detections overlapping to merge tracks; [20] enriches the predictions with egomotion computed from velocity and yaw sensors; and [72], in addition to Kalman filtering, uses Bayesian probability to provide certainty, trajectory and speed of pedestrians over time.

Particle filters are also widely used in tracking. Giebel et al. [69] use them to track multiple objects in 3D (in this case the cues are silhouette, texture and stereo). Philomin et al. [128] use the Condensation method [78] (a variant of particle filters) to track silhouettes approximated by B-Splines. Arndt et al. [5] employ particle filters in a track-before-detect paradigm, by coupling the tracking algorithm to a cascade classifier [159]. The realtime GPU implementation of particle filters by Mateo et al. [105] is also worth mentioning.

Recently, Leibe et al. [90] proposed the use of a color model and what they refer to as the *event cone*, that is, the space-time volume in which the trajectory of a tracked object is sought. The authors claim that although this proposal relies on the same equations as the Kalman filter, it is superior to it in the sense that object state estimation can be based on several previous steps, and multiple trajectories for the observed data can be evaluated.

Zhang et al. [170] propose the use of network flows to optimize association of detections to tracks. A min-cost flow algorithm is used to perform the detection-track association, and an explicit occlusion model is used to control long-term occlusions.

Research on detection in crowded scenarios has recently led to coupled detection-tracking frameworks, which share information between both modules, instead of treating them as independent stages. Gammeter et al. [63] perform multi-body tracking by combining the *implicit shape model* detector [92] and the stereo-odometry based tracker of [45]. Each trajectory is passed to a single-person articulated tracker, which estimates the 3D pose and dynamics of each individual. Andriluka et al. [3] detect targets using a part-based detector and then use a Gaussian process latent variable model to compute the temporal consistency of detections over time. Finally, Singh et al. [143] use the output from the part-based detector in [166] to initialize tracklets (short track segments of high confidence detections) and residuals (low confidence detections). The tracklet descriptors (based on color, motion and 3D height) and tracklet paths (using multiple hypotheses) are then associated within a global optimization framework.

#### Analysis

Tracking represents an important aspect for transforming a pedestrian detection algorithm into a PPS. However, this module has not received as much attention as other modules; each paper presents its own proposal and no comparisons have been made. Hence, it is not easy to extract conclusions. It can be said that the Kalman filter is by far the most heavily utilized algorithm, but tracking cues range from simple 2D ROI localization to color, silhouette, texture or 3D information. Recent coupled detection-tracking algorithms represent a promising way to exploit richer tracking cues, for example, tracking independently detected body parts instead of complete, rigid pedestrian silhouette models.

In our opinion, although there are many interesting proposals, much work remains to be done in tracking benchmarking before solid conclusions can be reached on this topic.

### 2.2.6 Application

The last module of a PPS takes high-level decisions based on the information from previous modules. In this case the role of Computer Vision research is not so relevant: psychology and human-machine interaction are the key areas here.

#### Review

This stage can be divided into two different subtasks: situation understanding and actions taking. The former one is focused on the particular driving situation by analyzing pedestrians behavior, studying the probabilities of collision, etc. It is aimed at predicting possible dangerous situations by making use of the data provided by the previous modules. The later one makes use of the prediction to take the *front-end* action to control counter measures, for example to warn the driver or the pedestrians, decelerate the car or even non-reversible active actions like the deployment of external airbags.

One first approach to infer basic pedestrian behavior is to estimate the walking direction of pedestrians [142], which is a potential cue to evaluate collision risk. Tsuji et al. [154] compute the relative moving vectors of pedestrians, which together with yaw and speed information of the vehicle is used to judge possible collisions. Fuerstenberg et al. [83] use laserscanners to predict the time to collision of pedestrians entering the so-called *region of no scape*, i.e., the region where a crash is unavoidable. These techniques, along with other surveillance-oriented pedestrian behavior analysis techniques are briefly reviewed in [64].

Usually, publications do not cover the later subtask part of the analysis, i.e., action taking. In [103] it is described how in the context of the SAVE-U project two types of actions are implemented at the application level: acoustical driver warning and automatic braking. These counter measures are applied with respect to a strategy based on three phases:

- *Phase 1: Early detection.* The system detects and tracks all pedestrians in front of the vehicle (within the sensor coverage area), but none of the protection measures are activated yet.

- *Phase 2: Acoustical driver warning.* A pedestrian is detected to enter the vehicle's path, but there is no risk of an immediate collision yet. The driver is alerted by an acoustical signal about this potentially dangerous situation.

- *Phase 3: Automatic braking.* A high risk of a collision has been identified. The vehicle is decelerated in order to avert the collision or, in the case that the collision is unavoidable, mitigate the impact.

Given both the difficulty of estimating the speed of pedestrians and their unpredictable behavior (e.g., suddenly start or stop braking, even change direction) the system takes decisions only based on pedestrian position and direction, but not speed.

In [154], Tsuji et al. follow a different approach. In this case, the authors make tests with two configurations for nighttime scenarios. The first one consists of a *head-up-display* (HUD) that projects the acquired TIR images in the windshield (just above the steering wheel in front of the driver) plus an additional voice assistance (no details are given about it). The other configuration consists of a LCD screen mounted over the gearbox together with the voice assistance. Tests highlight that the first configuration is more effective, aiding the driver to initiate collision avoidance in about one second. It is worth to note that in this case the final decision is left to the driver, so no automatic actions are applied on the vehicle.

Graf et al. [71] also propose two human machine interfaces for night driving. The first one consists in a small display on top of the steering wheel which just turns on when pedestrians are present in the road, fading out when there is no danger. The second does not require the driver to focus his/her attention on a display since it consists in a bar of LEDS that turn on when pedestrians are detected.

### Analysis

There are many functionalities that can be implemented at the application level: automatic braking, acoustical warnings, enhanced image visualizations (e.g., TIR image on HUD), evasive steering (with knowledge of the environment), external airbags, warning hoot for pedestrians, etc. However, even though the application level is based on almost abstract and filtered information, the functionality implementation is far from being trivial. For instance, warning signals shall be generated early to let the driver react properly; however, since pedestrians have a very high degree of dynamic freedom they can change their moving direction in a second, making it difficult to choose the precise time to deploy the warning. Another example is the true positive / false alarms rate. Too many false alarms could make the driver distrust the system. On the contrary, drivers could unconsciously reduce their attention if they delegate too much responsibility to the system. In this sense, high level reasoning combining systems looking inside the car (e.g., driver monitoring [9]) with the forward-looking proposals (like PPSs) seems to be the option to follow. For example, when the driver is aware of what happens in the scene, the so-called *driver in the loop*, the system should not deploy any warning.

### 2.2.7 Sensors and fusion

As previously stated, all of the reviewed techniques rely on the output of cameras. They are the most widely used sensors, due to the high potential of visual features, high spatial resolution and richness of texture and color cues. However, it is clear throughout the review that image analysis is far from simple: cluttering and illumination, among many other factors, do affect performance. Furthermore, the VS can be affected by glaring sources of light, while TIR can be influenced by other *hot* objects (e.g., engines of other vehicles or light poles), changing weather conditions (i.e., relative temperature changes), year/season, etc [31]. In fact, pedestrians could be warmer or colder than the background, depending on such factors [70].

Fusion of VS/TIR sensors and active sensors, which are used to obtain complementary information, is being investigated in the context of on-board pedestrian detection. The strengths and weaknesses of different kinds of sensors can be complemented in order to improve the overall system performance. Active sensors are based on technologies that emit signals and observe their reflection from the objects in the environment, for example radars emitting radio waves or laserscanners emitting infrared light. In general, these sensors are convenient for detecting objects and providing superior range estimates out to larger distances relative to passive sensors.

Next we review some systems that implement sensor fusion. Table C.4 in App. C provides a summary of the most relevant systems.

#### Review

Fardi et al. [50] combine a laserscanner with a TIR shape extraction method to select ROIs, using Kalman filtering as the data fusion algorithm. Premebida et al. [130] segment and track clusters of points along the 1D laserscanner dimension (note that tracking and segmentation are performed together), while classification is performed using data from the laserscanner (using a Gaussian mixture to model clusters centroid, standard deviation, radius, etc.) and the VS (using Haar features and AdaBoost). Milch et al. [110] make use of radar, velocity and steering sensors to generate hypotheses. They then perform classification using a shape model for either the VS or the TIR spectrum images. Linzmeier et al. [98] also exploit radar, but combine it with thermopile, steering angle and ambient temperature sensors. In this case fusion can be done at a low level (candidate generation combining radar and thermopile) and at a high level (candidates independently generated by all the sensors).

Combining VS and TIR spectra from the two camera types has also been proposed. In [17], by Bertozzi et al., v-disparity is computed using VS, and then foreground segmentation is carried out in both the VS and TIR (2D area overlapping and 3D information are the fusion cues). Finally, symmetry and template matching are used to classify, verify and refine final detections in the TIR. Krotosky et al. [87] evaluate a tetra- and tri-sensor systems that utilize both the VS and TIR. For example, in the tri-sensor approach, a VS stereo pair performs ROI generation while VS, TIR and disparity-based HOG-like features are fed to an SVM for classification.

One of the systems of the SAVE-U project [103] attaches a radar sensor to the VS and TIR cameras. They implement three different levels of fusion: sensor-, low- and high-level. The first level is aimed at associating different radar detections (each from an independent sensor) into unique real objects, as well as establishing a correspondence between the VS and TIR images. For low level fusion, ROIs are first selected in the VS by an algorithm based on histogram edge orientations, then resized by radar data (i.e., they are adjusted to the ground by using the accurate radar distance estimation), and finally classified by NN-LRF (Sect. 2.2.3). High-level fusion associates the radar and VS information (distance and azimuth) of each object, and then tracks their trajectories.

### Analysis

Sensor fusion for ADAS is an open area of research, and much much work is still required before convincing results are achieved in real scenarios. The ideal combination of sensors shall be clarified, given that each sensor has its own failure cases. For example, the conclusions of the SAVE-U project [103] state that although the radar-camera combination works well in simple test tracks the radar becomes unreliable at 10-15 m when working in real scenes, due to reflections from other objects (humans have low reflectance). Laserscanner, which work with infrared beams, are progressively gaining the interest of researchers as they can detect pedestrians while providing accurate distance estimates. However, laserscanners are affected by adverse weather conditions like cameras, which is not the case for radar.

## 2.3 Benchmarking

In contrast to other areas, like face detection or document analysis, pedestrian detection for ADAS lacks of well-established datasets and benchmarking protocols. The absence of realistic public datasets and the difficulty of implementing published techniques have usually led researchers to evaluate new proposals with local private datasets without any comparison to other state-of-the-art proposals. Public datasets are necessary for two reasons: 1) to evaluate algorithms with different example sets, taken at different places under different conditions, but specifically from different research groups (which adds extra variability); and 2) to compare new algorithms with existing ones, that is, given that it is hard to reproduce algorithms, the easiest way of establishing comparisons is to compare results from the same datasets following the same criteria.

There are some specific requirements that a pedestrian dataset shall fulfill to be specifically used in PPSs. Some of them are a must in any set while others make easier the task of evaluating different aspects of classifiers. They can be summarized in the following points:

- Topic significance. This first one may seem obvious, but it is important that the test data is the most similar to the final application as possible in this case ADAS environments. This means that pedestrians must be standing approximately in the same plane as the on-board camera placed at a realistic height from the ground.
- Quantity. Given the variability of the target, the number of examples shall be high, for example, at least $1,000$ positive samples for training.
- Resolution. As has been seen, the range of pedestrians sizes in the image is large due to perspective and distance. Given that algorithms can either make use of the resized or the original size (depending on the classifier) it is desirable to make both approaches available. By providing this data researchers get a well defined set in both cases avoiding to have to reconstruct these cases.
- Sequences. Cropped samples are useful for the object classification module, but in order to benchmark the whole system full annotated video sequences are required.

The first pedestrian dataset is probably the MIT Pedestrian Database [124], made public in year 2000. In 2005, Dalal et al. (INRIA) [35] present one of the most widely known datasets, consisting in a large number of samples containing persons in many different poses and situations. This dataset is not related to ADAS but contains high quality images taken from photographs and from the Internet, so their quality tends to be high (pedestrians are at least 128 pixels high) and the variety of scenarios ranges from typical city streets to mountain landscapes. However, in the recent years it has become as the standard benchmark for any new human detector. Daimler Pedestrian Classification Benchmark (DC-01) [114] and the Computer Vision Center Pedestrian Dataset 01 [68] are the first specifically ADAS oriented pedestrian datasets, containing images taken from cameras mounted on a vehicle. The samples are significantly smaller (36 and 24 pixels high, respectively), all taken from street scenarios, so in contrast with INRIA, so in this case there are not out-of-the-topic images. Between 2007 and 2008 there appeared some papers presenting datasets to support the novel techniques (USC [166, 166], ETH [46] and TUD [3]) and a paper proposing a new dataset by NICTA [122].

From all these sets, just DC-01, CVC-01 and NICTA are ADAS-specific, so although the others can already provide an intuition on detection algorithms performance, just the former provide relevant statistics for ADAS. As can be seen in Table 2.2, all the aforementioned sets contain cropped images or, in the best case, isolated frames. Although this is sometimes sufficient to evaluate the object classification module performance, full video sequences are required to train and evaluate the other modules (e.g., tracking or verification) and the whole system. Between 2008 and 2009 two new pedestrian datasets have been presented aimed at resolving this lacks. The first one is the Caltech Pedestrian Dataset [41], which contains on-board video sequences containing instance annotation. Although this dataset seems promising at a first glance given the spectacular number of samples stated, the number of single pedestrians is similar to the previous ones, and are not so precisely annotated as for example NICTA. Moreover, the authors do not provide testing data, which represents a big inconvenience for researchers. The second dataset is Daimler Pedestrian Detection Benchmark (DC-02) [43], which contains grayscale resized training examples and fully annotated video sequences. Figure 2.9 illustrates some examples of these datasets.

In this thesis we present the Computer Vision Center Pedestrian Dataset 02 (CVC-02), which consists of three data subsets each one devoted to a specific task in pedestrian detection: candidate generation (CVC-02-CG), pedestrian classification (CVC-02-Classification) and pedestrian detection system (CVC-02-System). Up to our knowledge, CVC-02 is pioneer in several aspects:

- CVC-02-CG is the first specific candidate generation evaluation dataset.
- We provide original sized and resized cropped examples for CVC-02-Classification.
- We provide both cropped negative sets, coming either from sliding window candidates or from the road, and person-free frames in CVC-02-Classification.
- We provide both color and depth maps when possible, which we expect to be one of the exploited cues for classification in the future research.

More details of the proposed dataset are given in App. B. In Table 2.2, following the pedestrian datasets summary by Dollár et al. in [41], we provide detailed information of the state of the art datasets.

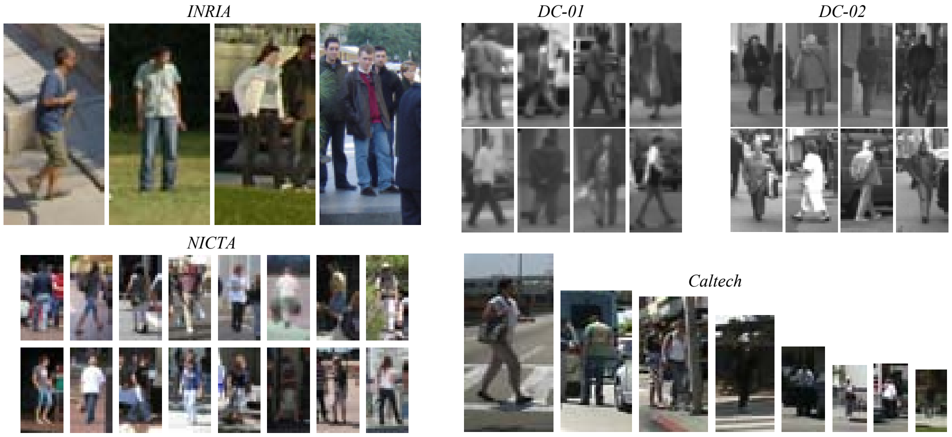


**Figure 2.9:** Positive examples from different pedestrian datasets.

**Table 2.2:** Summary of the current pedestrian datasets, following the idea by Dollár et al. [41]. The bottom datasets are the most convenient for evaluation at the moment. We state raw annotations, many of the datasets make use of mirroring ([35, 68, 114] and jittering [114]. Notes: †From all the annotations, we found that just around a 5% can be termed as useful and well annotated. ‡Test data is not publicly available.

| | Train | | | Test (window) | | | Test (image) | | Properties | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | #pedestrians | #negatives | #negative images | #pedestrians | #negatives | #negative images | #pedestrians | #frames | ADAS | Color | Original Scale | Video | Instance Annot. | Year |
| INRIA [35] | 1 208 | – | 1 218 | 566 | – | 453 | 566 | 291 | – | ✓ | ✓ | – | – | 2005 |
| CVC-01 [68] | 700 | 4 000 | – | 300 | 1 000 | – | – | – | ✓ | ✓ | ✓ | – | – | 2006 |
| DC-01 [114] | 2 400 | 5 000 | 15 000 | 1 600 | 10 000 | – | – | – | ✓ | – | – | – | – | 2006 |
| USC [166, 165] | – | – | – | 816 | – | – | – | – | – | – | – | – | – | 2007 |
| ETH [46] | 2 388 | – | 499 | – | – | – | 12 000 | 1 804 | – | ✓ | ✓ | ✓ | – | 2007 |
| TUD [3] | 400 | – | – | – | – | – | 311 | 250 | – | ✓ | ✓ | ✓ | – | 2008 |
| NICTA [122] | 37 344 | 200 000 | 4 147 | 6 879 | 50 000 | 1 004 | – | – | ✓ | ✓ | – | – | – | 2008 |
| DC-02 [43] | 3 915 | – | 6 744 | – | – | – | 56 492 | 21 790 | ✓ | – | – | ✓ | ✓ | 2008 |
| Caltech [41] | 120 000† | – | 61 000 | – | – | – | ‡ | ‡ | ✓ | ✓ | ✓ | ✓ | ✓ | 2009 |
| CVC-02-CG | – | – | – | – | – | – | 266 | 100 | ✓ | ✓ | ✓ | – | – | 2009 |
| CVC-02-Class. | 1 016 | 7 650 | 153 | 570 | 7 500 | 150 | 587 | 250 | ✓ | ✓ | ✓ | – | – | 2009 |
| CVC-02-Sys. | – | – | – | – | – | – | 7 983 | 4 634 | ✓ | ✓ | ✓ | ✓ | – | 2009 |

## 2.4 Discussion

A perfect on-board PPS must detect the presence of people in the way of the vehicle and react according to the risk of running over the pedestrian (warn the driver, brake the vehicle, deploy external airbags, perform an evasive maneuver), without disturbing the driver if there is no risk at all. Moreover, such a system should work well independently of the time, road and weather condition. Additionally, the cost of the pedestrian detection system should be relatively small compared to the total cost of the vehicle.

It is clear in the reviewed literature that an enormous research effort has been made in automatic people detection during the last decade. However, despite of the significant advances in the area we are not even halfway to an ideal PPS. Indeed, from our opinion, there are some open issues not researched in the literature that are of crucial importance for the area:

- Foreground segmentation has been poorly researched. Most of the papers just focus on classification, and the ones that include any kind of candidate selection algorithm explain it poorly and provide few details and evaluations on this specific task. In fact, there does not exist any performance evaluation protocol or dataset for this task.

- With respect to the previous point, the performance of the different independent tasks (i.e., the modules presented in this chapter) has also been ommited in the literature. A study on the different parts and their effects on the later stages of the system would be of great interest.

- Pedestrian refinement in general also needs to be further explored. The literature on window candidates clustering is almost non existing, so any study on this point would be of interest. In addition, the relation between the clustering and the other actors of the detection should also be studied (e.g., how the behavior of classifier affects the clustering).

- New ADAS-specific pedestrian datasets are needed. It is clear that the well-known INRIA dataset is useful to have an idea of how classification algorithms work on generic human retrieval. However, PPSs need specific datasets to perform experiments with the most similar data that the systems will find in the real applications. Recent datasets (Daimler and Caltech) point to this direction, but there is still much way forward aspects like datasets quantity (e.g., face detection can be evaluated using more than twenty datasets), representativity (children or tall people are typically out of PSC), variability (height, distance, degree of occlusion, complexity of background, clothes, pose, illumination, etc.) and resolution (including far pedestrians), multimodal data (including depth either from stereo or active sensors).

- Sensors. Until now we have referred to visible spectrum datasets, but if we look for a public TIR ADAS-oriented datasets we will not find one. To the best of our knowledge, the OTCBVS [38] is the only publicly available TIR dataset, but it is not suited to ADAS benchmarking. The same problem is found with active sensors, such as laserscanner or radar.

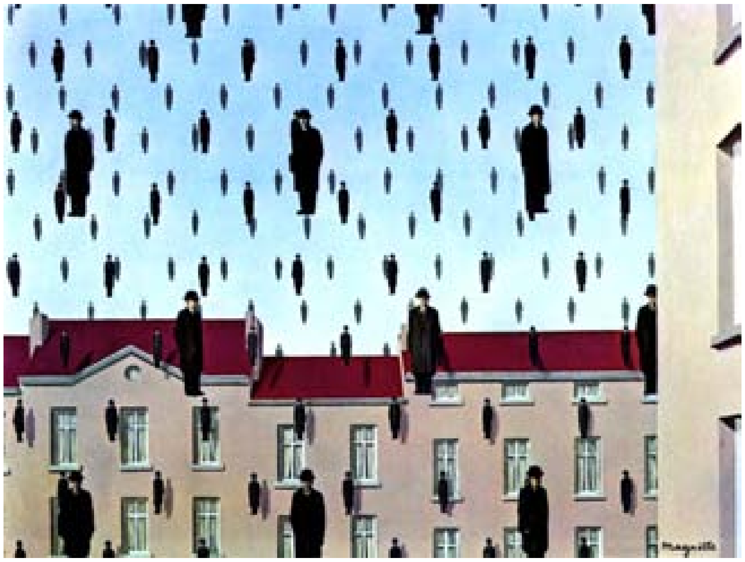
- Once we have stated the lack of datasets, we can go further and point to the most ommited piece of the data acquisition process in the literature: the annotation tools. The task of annotating independent samples, say a thousand, is boring and tiring. However, as datasets evolve and sequence annotation starts to be a need, the annotation then turns not only tiring but also imprecise, time consuming and unhealthy. New intelligent annotation techniques are needed, for instance including detection or unsupervised silhouette matching in the annotation task, temporal coherence, etc.

- Following the need of improving annotation techniques, pedestrian shape data require an special consideration. Annotating a pedestrian window requires a few mouse clicks (three in our case App. B), but it is not comparable to working with silhouettes. New techniques to perform fast silhouette annotation or even unsupervised shape extraction algorithms would be of interest for the community.

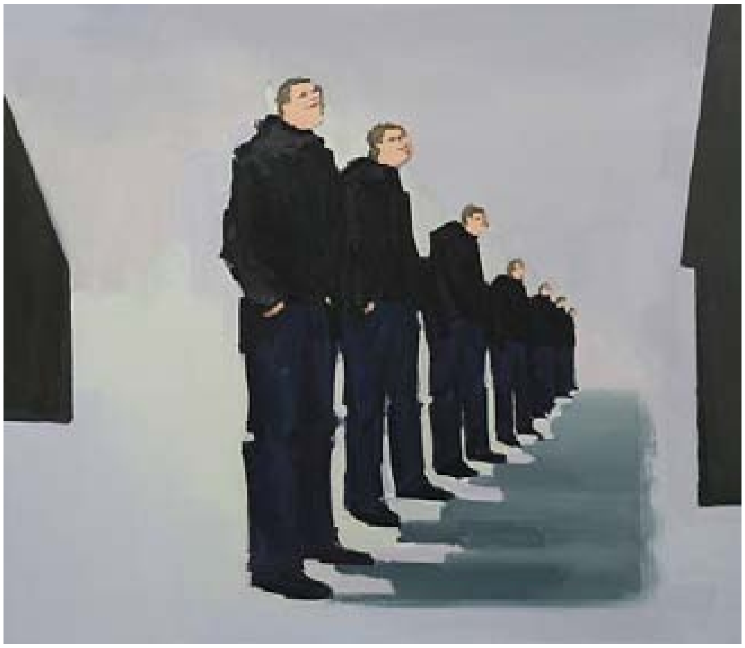
- As has been seen, there are many papers dealing with pedestrian detection on nighttime. However, they are unrelated with each other, perhaps as a result of the lack of common datasets, as has been commented. Detection in TIR imagery has its own problems: pedestrians are not the only hot object (e.g., vehicles, light poles or traffic signs heated by the sun are displayed too), the relative temperature during the day and along the year change, etc. According to this, although rough segmentation methods (understood as attention mechanisms) are probably easier to implement for TIR images, the classification task in general turns more difficult than in VS.

## 2.5 Summary

In this chapter we have presented a comprehensive survey of PPSs. Although the area of research is relatively young, the literature can be described as big and chaotic. We have made use of the proposed general architecture to review the existing techniques according to their place in the architecture pipeline and then analyze them. In addition, we include a review on systems making use of sensor fusion techniques, which points as a promising future research line, and include an overview of benchmarking aspects. Finally, we provide discussion on the most relevant topics and point out some interesting lines of research.



*Golconda*
Renné Magritte, oil on canvas, 1953
(Menil Collection, Houston, United States)



*Exposition (line)*
Andrius Zakarauskas, oil on canvas, 2007
(Lithuatian Art Museum, Vilnius, Lithuania)

*Golconda* and *Exposition (line)* represent the opposite approaches presented in this chapter. The former represents a city with men in overcoats and hats distributed all over the scene, challenging any physical rule. In fact, they are placed in a regular hexagonal grid following a scale progression, which is a common scanning procedure in pedestrian detection. The author tried to insinuate the monotonous nature of human existence, consisting in repeatedly doing the same things as an automata, which has parallelism with a typical blind-based algorithm studied in this chapter. In addition, the author also expresses how representations of things, in opposition to things themselves, can lie: a real person cannot be placed in all the possible places of a scenario. The latter painting depicts a scene in which the figure of a man is placed in a line following the perspective, which provides a more sensible and adjusted to reality scenario. Along the chapter these two opposite viewpoints are contrasted with each other. One important aspect to point out with respect to *Exposition (line)*, but that it is applicable to the problem of object detection, is the gradual distortion of objects with respect to scale. The author smartly illustrates the problematic between figurative and abstract representations of reality, which in fact are part of the same scene after image formation in the form of near and far objects, respectively.

# Chapter 3

# Foreground segmentation

Foreground segmentation (or candidate generation in the case that it extracts specific windows), has not received as much attention as other tasks involved in pedestrian detection. PPSs are still a new area of research so most of the works have been addressed to perhaps the most attractive of the tasks: object classification. However, as will be seen along this chapter, the choose of the appropiate techniques for the foreground segmentation is of paramount importance for real-world intelligent vehicles applications.

In Chap. 2, we define the main objective of foreground segmentation as the extraction of ROIs from the input image to be classified, avoiding as many background regions as possible. This latter requirement has two potential benefits for pedestrian detection, which are in fact the final motto's of foreground segmentation. First, given that a perfect zero false-positive-rate classifier does not exist, if the ROIs not containing pedestrians is reduced, then it can be expected that the number of false positives of the whole system will also decrease. Second, if these potential false positives are discarded in this early stage, the fewer the ROIs will have to be classified, hence the computation time required in the whole system will decrease.

In this chapter we present a comprehensive study of candidate generation techniques to be used in the foreground segmentation. We address candidate generation instead of other foreground segmentation approaches like the implicit shape model [92] or Chamfer System [67] by attending to two reasons: windows generation is the most used approach in the literature and are the most suitable approach to cover the range from 0 to 50 m. Given the reduced literature addressing this area, the study is not just constrained to the comparison of different existing techniques. Instead, a more global approach is first set in Sect. 3.1 by establishing a groundtruth dataset and the protocols that are later used to test and compare the performance of the algorithms. Then, in Sections 3.3.1 and 3.3.2 we formalize two of the existing techniques used for candidate generation, sliding window and flat-world-assumption, which will serve as baseline to compare our proposed techniques. The first represents the most simple candidate generation technique, in fact it can barely be considered as a foreground segmentation technique since it selects *every possible window* in the image. The latter makes use of prior information of the scene to reduce the number of windows to be processed, and reduce the computation time accordingly. Both techniques can be typically used for monocular cameras. However, as stereo rigs are gaining popularity, PPSs have started to exploit their properties. In fact, as previously highlighted, stereo segmentation stands as one of the most used approaches in PPSs. Given the problems found in these techniques, in Sections 3.3.4, 3.3.5 and 3.3.6 we present three new techniques aiming at fulfill the objectives of foreground segmentation: adaptive road scanning, adaptive road scanning with 3D-based filtering, and probabilistic multi-cue 3D scanning. Finally, a global comparison of all the described algorithms together with a final discussion are made in Sections 3.4 and 3.5, with the aim of setting the state-of-the-art in this area.

## 3.1 Evaluation methodology

Contrary to other research areas such as object classification, in which many protocols are available to compare the methods performance, candidate generation lacks from well-established ones. The measures to compare the performance of different object classification methods are typically based on ratios between correct and incorrect classification of positive and negative samples. For example, a ROC curve (App. D) relates the number of correctly classified positives and the misclassified negatives. However, in the case of candidate generation not all these measures are suitable since the objectives are different. According to this, we propose a new evaluation protocol based on the objectives, which are described as follows:

($o_1$) to minimize the number of non pedestrian candidates (NPC), i.e., potential false positives of the classification;

($o_2$) to minimize the number of annotations not selected as candidates, i.e., potential false negatives (FN), which is related to the potential true positive (TP) that marks an annotation as a selected candidate;

($o_3$) to maximize the number of candidates per annotation (CPA), i.e., the larger CPA is, the more pedestrians are likely to be selected by the classifier and the richer will be the input for an hypothetical non-maximum-suppression algorithm. Note that this measure is complementary to TP and FN in the sense that these ones do not take into account multiple candidates per annotation (see Fig. 3.1).

In order to classify a window as NPC, FN or TP, we make use of the overlap measure between groundtruth and detection proposed by Everingham et al. [47] for object detection evaluation in the PASCAL Challenge [126]. A candidate window $c$ is marked as TP if its overlap with an annotated window $a$ exceeds a certain threshold $\Gamma$, where

$$\text{overlap}(c, a) = \frac{\text{area}(a \cap c)}{\text{area}(a \cup c)} \ , \tag{3.1}$$

otherwise $c$ is marked as NPC. Figure 3.1($a$) illustrates the overlap criterion. In addition, when a $a$ does not have any associated candidate, it is marked as a FN. In the case of [47], $\Gamma = 0.5$. In our case, given that we are selecting windows to be later processed, a TP candidate must frame the annotation in a way that the later classification ideally classifies it as positive, so such a low threshold is useless. Some qualitative experiments with different thresholds and some basic candidate generation algorithms let us define the appropriate threshold as $\Gamma = 0.85$. Figure 3.2($b$) illustrates some examples of candidate windows marked as TP, stating their corresponding overlap measure with the annotation. As can be seen, there are candidates that do not frame perfectly the pedestrian, such as the last one with an overlap of 0.9. One can think that $\Gamma = 0.85$ is a too relaxed threshold, however, two aspects make us think that 0.85 is an adequate value. First, an object classifier must be able to deal with not-perfectly-aligned windows to the targets (in fact, many times there is not such a unique correct alignment). Second, the simple sliding window algorithm is supposed to provide many candidates for single pedestrians, to be later reduced to one by a refinement technique. If $\Gamma$ is increased, even such an exhaustive algorithm suffers a significant performance decrease, which in fact is not real.

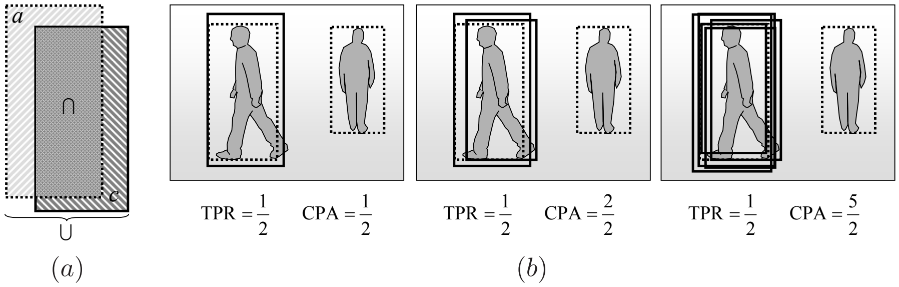


**Figure 3.1:** ($a$) Overlapping Criterion: $c$ represents the candidate window and $a$ the annotated window. ($b$) Difference between True Positive Rate (TPR) and Candidates Per Annotation (CPA). CPA increases with the number of overlapping windows whereas TPR measures if any candidate overlaps an annotation, no matter the number.

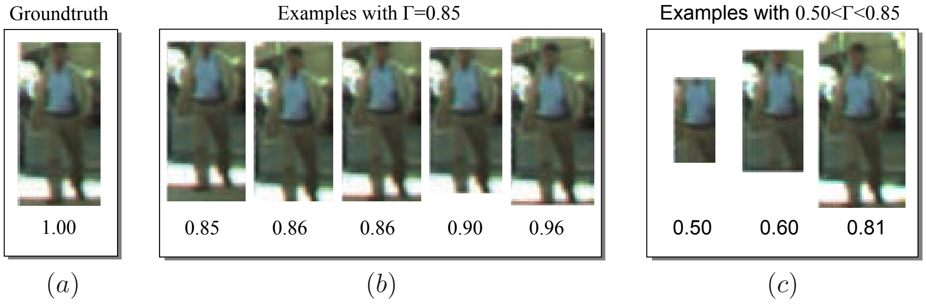


**Figure 3.2:** ($a$) Groundtruth. ($b$) Examples of candidates marked TP attending to the overlap criterion with $\Gamma > 0.85$ and ($c$) with $0.50 < \Gamma < 0.85$.

Once the criterion to classify a candidate as pedestrian or non-pedestrian is defined, the next step is to describe both the performance measures and their visual representation. Attending to the objectives of the algorithm, we can analyze the suitability (marked with a ✓) of the measures from the existing evaluation protocols:

✓ True Positive Rate ≡ Sensitivity ≡ Recall = $\frac{TP}{TP+FN}$, in short TPR, relates the number of annotations selected as a candidate (i.e., the overlap criterion with the annotation is fulfilled for at least one candidate) with the total number of annotations. It is useful to represent the aforementioned objective ($o_2$).

× False Positive Rate = $\frac{FP}{FP+FN}$, in short FPR, relates the misclassified negatives with the total number of negatives. In this case, it would translate into a ratio between the selected and the potential non pedestrian candidates (NPC). Therefore, this measure is useless since the potential NPC is not well-defined.

× Specificity = $\frac{TN}{TN+FP}$ is equivalent to FPR, so it is neither applicable in our case.

× Precision = $\frac{TP}{TP+FP}$ relates the TP with the number of generated candidates, which in our case is the sum of TP and NPC. This kind of measure is not suitable because $NPC >> TP$, e.g., NPC is in the order of $10^4$ to $10^6$ and TP is less than 10.

As can be seen, objectives ($o_1$) and ($o_3$) can not be expressed using the current measures, so two specific measures shall be used for them. In the case of ($o_1$), we think that expressing the NPC as an absolute value is more informative than using a ratio (e.g., precision), since it is relevant by itself. In addition, choosing an arbitrary upper limit of NPC would result in a meaningless measure for this task. In the case of ($o_3$), although the use of a ratio is feasible for CPA, for example by taking into account the total number of $c$ that would fulfill overlap$(c, a) > \Gamma$ for a given $a$, there are two reasons that made us prefer the absolute value. First, the possible number of $c$ depends on the size of the annotation (e.g., if $\Gamma = 0.85$, there are around $1,000$ for a $64 \times 128$ annotation and not more than 10 for a $12 \times 24$), so the use of a ratio would not be informative at all. Second, although the CPA value must be high, there shall be a limit in order not to over scan the image, that is, $1,000$ candidates for a single annotation is not the pursued value.

Accordingly, a plot should represent such measures in a clear way such that an expert could answer a set of performance evaluation questions by analyzing it. Let us define some important questions as: What is the representation of a perfect candidate generation algorithm? How does the plot change if an algorithm is optimized to provide less FP? Is there such a dominant axis that should be prioritized when performing algorithm improvements?

We present the candidate generation performance (CGP) plot in Fig. 3.3, which provides answers for the aforementioned questions at a glance by plotting TPR, NPC and CPA. Contrary to other curves such as ROC or PR, in the case of CGP there is not such a perfect point since two of the axes represent absolute values and not ratios. Hence, the goal of a foreground segmentation algorithm can be defined as optimizing the tradeoff between the number of windows and the selected pedestrians, marked as *good algorithm* in the figure.

Along the chapter we will use this plot to measure the performance of the different algorithms. In each of them, we will use an adjusting parameter to explore and select the best configuration for each algorithm. Finally, a global evaluation will be made with the optimum parameters for each plot.

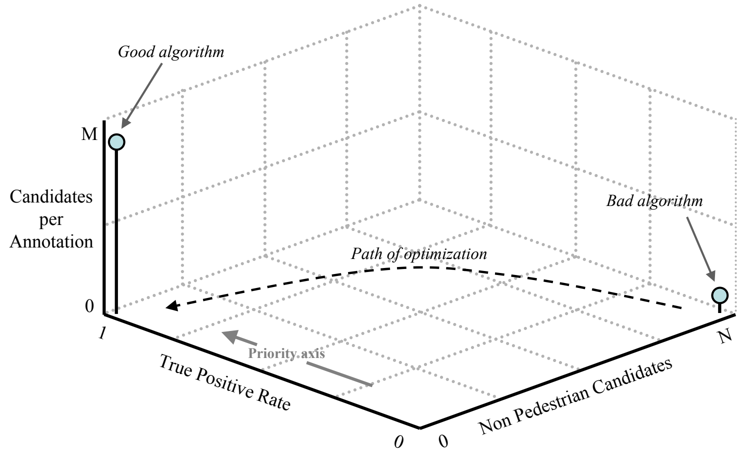


**Figure 3.3:** Scheme of the proposed candidate generation performance plot.

We make use of the proposed pedestrian candidate generation dataset (CVC-02-CG) detailed in App. B. It consists of 100 annotated random images from urban scenarios. In order to assure that the proposed algorithms are versatile and adaptable to real world conditions, the camera position is not necessarily the same in all of the frames. In fact, the relative road-camera position is likely to vary from frame to frame, as will be seen later.

## 3.2 Preliminary definitions

In this section we introduce the parameters that will be used by the described algorithms along the chapter. The generic input parameters for all the foreground segmentation algorithms are the image $I$, of width $I_w$ and height $I_h$; a minimum window $d^m$, of width $d^m_w$ and height $d^m_h$; and a maximum window $d^M$, of width $d^M_w$ and height $d^M_h$. In the case of pedestrians, the windows maintain a $x/y = 1/2$ aspect ratio. Other parameters used in the algorithms are the image center coordinates $(x_0, y_0)$ and the camera focal in both axes $(f_x, f_y)$, in pixels. The output of the algorithms is a set of candidates $\mathcal{C}$ in the image that will be sent to the classifier.

## 3.3 Algorithms

### 3.3.1 Sliding window

The simplest candidate generation method is the sliding window [124, 34], also referred to as exhaustive scan [68]. It consists in scanning the input image with windows of different scales at all the possible positions. One of the well-known implementations of the algorithm is the proposed by Dalal et al. [34], in which a scale pyramid is first constructed and then a fixed window of 64 × 128 pixels is used to scan each one of the scales. For the sake of comparison with the other algorithms, in our case we scale the windows instead of scaling the image. Figure 3.3.1 illustrates the algorithm.

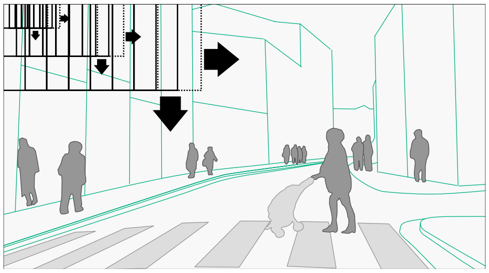


**Figure 3.4:** Scheme of the sliding window algorithm.

#### Algorithm

The parameters of the algorithm consist of the sliding position steps $(\Delta_x, \Delta_y)$ referred to the minimum window $d^m$ and the scale step parameter $\Delta_s$, used to define the different scales of the window. Algorithm 1 formalizes the technique.

**Algorithm 1** Sliding Window

```
Input:
  Spatial $\Delta_x$, $\Delta_y$ and scale $\Delta_s$ stride
Initialization:
  $S_1 = 1$ the smallest scale
  $S_N = \min(I_w/d_w^m, I_h/d_h^m)$ is the largest scale
  $N = \left\lfloor \frac{\log(S_N/S_1)}{\log(\Delta_s)} + 1 \right\rfloor$ is the number of scales
Algorithm:
  For each $s_i \in \{S_1, \dots, S_N\}$
    Compute parameters for $s_i$
      $(c_{xi}, c_{yi}) \leftarrow$ Proportional window size
      $(\Delta_{x_i}, \Delta_{y_i}) \leftarrow$ Proportional step
    Scan the image in the $X_I$,$Y_I$ axes using $c_{xi}$,$c_{yi}$
      Add new candidate $c_i$ to $\mathcal{C}$
```

### Performance evaluation

Although the algorithm may seem quite straight-forward, the task of choosing the parameters needed to achieve a satisfactory performance is not trivial. As can be seen in Table 3.1 and Fig. 3.5, the algorithm performance greatly varies depending on the chosen parameters. The finner the spatial and scale steps the higher the TPR, but also the larger the NPC. On the contrary, if coarser steps are chosen, NPC is reduced but TPR significantly decreases.

The parameters required to obtain a perfect TPR are $\Delta_s = 1.05$ pixels and $(\Delta_x, \Delta_y) = 1.5$ pixels, which in fact correspond to the ones used in [34] when mapped to a minimum window of $12 \times 24$ pixels. It guarantees a $TPR = 1$, i.e., zero false negatives for all the testing samples, at the cost of generating more than one million candidates.

**Table 3.1:** Number of windows generated by Sliding Window algorithm using different parameter configurations.

| | | Scale Step ($\Delta_s$) | | | |
|---|---|---|---|---|---|
| | | 1.05 | 1.10 | 1.15 | 1.20 |
| Spatial Step ($\Delta_x, \Delta_y$) | 1.0 | 2 902 605 | 1 559 409 | 1 112 592 | 890 307 |
| | 1.5 | 1 292 294 | 693 848 | 495 574 | 396 521 |
| | 2.0 | 728 523 | 391 604 | 279 330 | 223 297 |
| | 2.5 | 467 038 | 250 397 | 178 786 | 143 001 |
| | 3.0 | 325 032 | 174 701 | 124 705 | 99 753 |

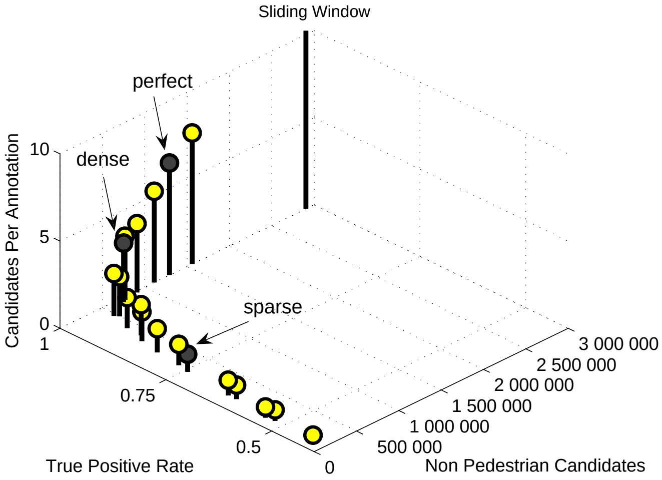
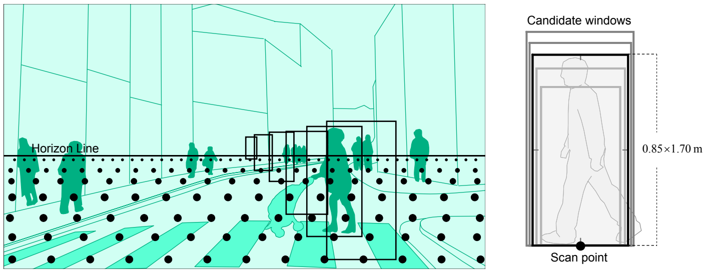


**Figure 3.5:** Performance plot of the sliding window algorithm. The shaded circles show three different significant configurations of ($\Delta_x$, $\Delta_y$, $\Delta_s$): perfect (1.5, 1.5, 1.05), dense (1.5, 1.5, 1.10) and sparse (2.0, 2.0, 1.2).

### 3.3.2 Flat world assumption

One million candidates to be classified will turn out in a high computation time during classification. Moreover, the potential number false positives is big, and since it is based on brute force it does not make use of any prior knowledge of such an application-focused system. Some ADAS researchers soon realized that a more sensible candidate generation approach could optimize the system performance.

Ponsa et al. [129] for vehicle detection, Gavrila et al. [66] and Broggi et al. [13] in the case of pedestrians, have made use of the so-called flat-world-assumption. The technique is based on the assumption that pedestrians are on the ground in front of the host vehicle, and that the geometry of the road and its position with respect to the camera does not change. According to these premises, the algorithm scans the road plane with pedestrian-sized windows, thus reducing the searching space to just the windows located onto an hypothetic road plane. Figure 3.6($a$) illustrates how the algorithm works.



(a) (b)

**Figure 3.6:** ($a$) The FWA algorithm scans the fixed ground plane. The horizon line is commonly used as a measure of the road position, given that it corresponds to the image projection of road points at infinity. ($b$) For each scan point, $\mathcal{S}$ candidate windows of different sizes are generated.

#### Algorithm

Let us define the world coordinate system $\mathbf{W} = (X_W, Y_W, Z_W)$ in such a way that the $X_W Z_W$ plane is contained in the road surface, just under the camera coordinate system $\mathbf{C} = (X_C, Y_C, Z_C)$; the $Y_W$ axis contains the origin of the camera coordinate system; the $X_W Y_W$ plane contains the $X_C$ axis and the $Z_W Y_W$ plane contains the $Z_C$ axis. Due to these constraints, the six extrinsic parameters (three for the position and three orientation angles) that refer $\mathbf{C}$ to $\mathbf{W}$ are reduced to just three: the camera roll $\phi$, pitch $\theta$ and height $h$. Figure 3.7 illustrates the coordinate systems and their position with respect to the vehicle.

As was mentioned, the main assumption of this algorithm is that the road geometry and its relative position with the camera do not change, which in fact corresponds to fixing the parameters $\phi$, $\theta$ and $h$. The additional parameters consist of the closest ground point seen from the camera[1] $Z_{W_{min}}$, and the furthest one $Z_{W_{max}}$; the range of scanning in the X axis $(X_{W_{min}}, X_{W_{max}})$; as well as the window stride $\Delta_x$ and $\Delta_z$ along the road X and Z axes, respectively.

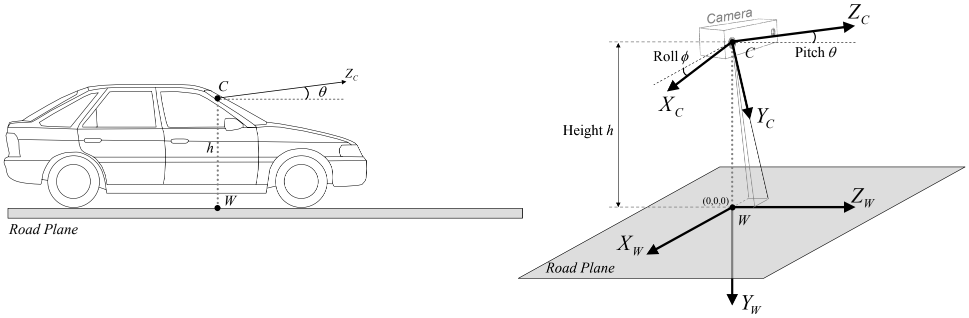


**Figure 3.7:** Relative camera-world coordinate systems. Fixing $\theta = 0°$ implies parallel $X_C$ and $X_W$ axes.

We assume a pedestrian to be $h = 1.70$ m high. In the case of the body width, a width margin is used to adjust most of human proportions and also leave some space for the extended extremities. Hence, the width is defined as a ratio of the height, specifically 1/2. For example, the mean pedestrian dimensions are 1.70 × 0.85 m, independently of the extra-margin added to the candidate by the classifier[2]. In order to cope with the variable dimensions of pedestrians, a standard deviation $\sigma = 0.2$ m is added to the candidates height, maintaining the corresponding aspect ratio when computing width. Hence, along the chapter we will use $\mathcal{S}$ different candidates for each scan point, as Fig. 3.6($b$) shows. Specifically, in our case we use $\mathcal{S} = 5$ (i.e., candidates of $(0.75\,m \times 1.50\,m)$, $(0.80\,m \times 1.60\,m)$, $(0.85\,m \times 1.70\,m)$, $(0.90\,m \times 1.80\,m)$, $(0.95\ m \times 1.90\ m)$), which has provided successful performance in the experiments.

For each candidate window $c$ with world coordinates $(c_X, c_Y, c_Z, c_W, c_H)$, where $(c_X, c_Y)$ are at the center of the window, its corresponding image coordinates ($c_x$, $c_y$, $c_w$, $c_h$) can be computed by making use of the projective equations and trigonometry:

$$c_w = f_x \cdot \frac{c_W}{c_Z} \ , \tag{3.2}$$

$$c_h = f_y \cdot \frac{c_H}{c_Z} \ , \tag{3.3}$$

$$c_y = x_0 + \frac{f_x \cdot c_X}{(h + c_Y) \cdot \sin(\theta) + c_Z \cdot \cos(\theta)} \ , \tag{3.4}$$

$$c_x = y_0 + \frac{f_y \cdot (h + c_Y) \cdot \sin(\theta) + c_Z \cdot \cos(\theta)}{(h + c_Y) \cdot \sin(\theta) + c_Z \cdot \cos(\theta)} - \frac{c_h}{2} \ . \tag{3.5}$$

[1] In our case, with a camera of 6 mm focal, oriented to the road avoiding to capture the vehicle hood, the first road point seen is at around 4 to 5 m (App. A).

[2] Dalal et al. [35] demonstrate that adding some margin to the window (33% in their case) results in a performance improvement in their classifier.

Notice that in Eq. 3.5 a factor is added to place the $y$ coordinate at the center of the candidate window.

The formalization of this method can be followed in Algorithm 2.

**Algorithm 2** Flat World Assumption

```
Input:
  $\theta \leftarrow$ Relative camera-road pitch angle
  $h \leftarrow$ Relative camera-road height
  $Z_{min}, Z_{max} \leftarrow$ closest and furthest road points in the Z axis
  $X_{min}, X_{max} \leftarrow$ scanning range in X axis
  $\Delta_x, \Delta_z \leftarrow$ scanning stride in both axes
  $\mathcal{S}$ different window sizes for each scanning point
Algorithm:
  For all the positions from $X_{W_{\text{min}}}$ to $X_{W\text{max}}$ using $\Delta_x$
    For all the positions from $Z_{W_{\text{min}}}$ to $Z_{W\text{max}}$ using $\Delta_z$
      For all the window sizes $\mathcal{S}$
        Project the window $c$ using $h$ and $\theta$ and Equations 3.2 and 3.3
        Insert new window $c$ to $\mathcal{C}$
```

### Performance evaluation

In order to select the algorithm parameters for our images, we have fixed the roll to $\phi = 0°$, which can be assumed constant, as will be explained later; the camera height is fixed to $h = 1.25$ m, although it can actually suffer variations from 1.10 to 1.30 m as a result of the pavement and the vehicle suspension; and tested different values of $\theta$ in the range $[-5°, +5°]$. The grid is configured with the parameters $Z_{W_{min}} = 3$ m, $Z_{W_{max}} = 50$ m, $X_{W_{min}} = -10$ m, $X_{W_{max}} = 10$ m, $\Delta_x = 0.075$ m, $\Delta_z = 0.5$ m. Figure 3.8 illustrates the algorithm performance. As can be seen, the TPR is low in general, independently from the $\theta$ value. Besides, in experiments that we will see later on we have noticed that reducing $\Delta_x$ and $\Delta_z$, i.e., increasing the grid points, does not affect the TPR but increases the number of NPC.

Figure 3.9 displays the results of the algorithm on two test images with $\theta = 1°$. As can be seen, in many cases the fixed plane does not coincide with the real one due to a slightly different camera mounting (different $\theta$) or a variable road slope from frame to frame, hence the candidate windows are not correct and the TPR drops.

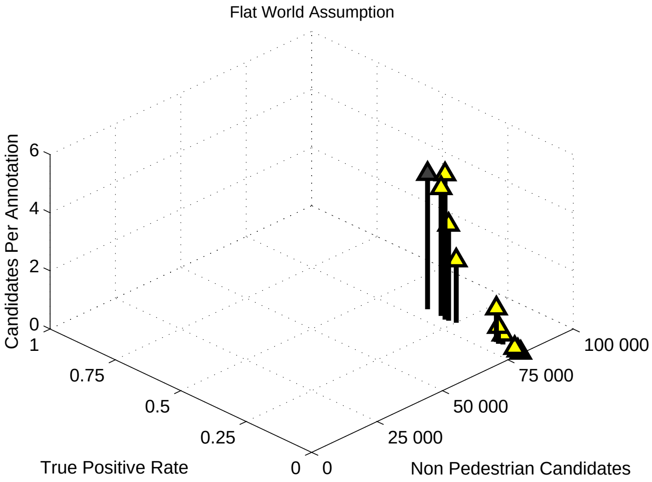


**Figure 3.8:** The shaded triangle represents the algorithm with $\theta = 1°$, the best configuration, which is chosen for the later algorithm comparisons.

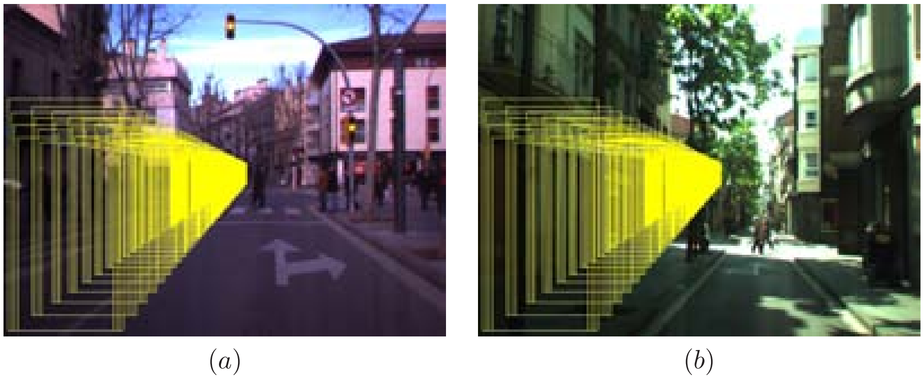


**Figure 3.9:** Results of the Flat World Assumption algorithm (in order to enhance the visualization just the row of $X_W = -3$ m is shown). $(a)$ The grid coincides with the real road plane, so all pedestrians are correctly framed. $(b)$ The estimated road plane does not match with the real one, so far pedestrians are not selected as candidates.

### 3.3.3 Stereo segmentation

Stereo imagery is a natural approach to object segmentation. For example, stereo is one of the main cues used to detect obstacles in the human vision system. Thus, at a first glance, stereo segmentation seems to be a promising method to generate candidates. However, from our experience, the single use of stereo is far from being useful for this task. There are many difficulties that make it a hard task. In Fig. 3.10 we illustrate some of these problems, which can be summarized in:

- Nonexistent 3D points (Fig. 3.10($b$)), which can lead to incorrect size of the candidate window.
- Overlapping close objects (Fig. 3.10($c$)), which occurs when two objects are very close to each other in $Z_C$. This is a typical situation with pedestrians appearing from behind a parked vehicle.
- Uniform regions (Fig. 3.10($d$)), which consist in pedestrians in front of buildings or street furniture, which makes it difficult to extract a reliable bounding box of the 3D object.

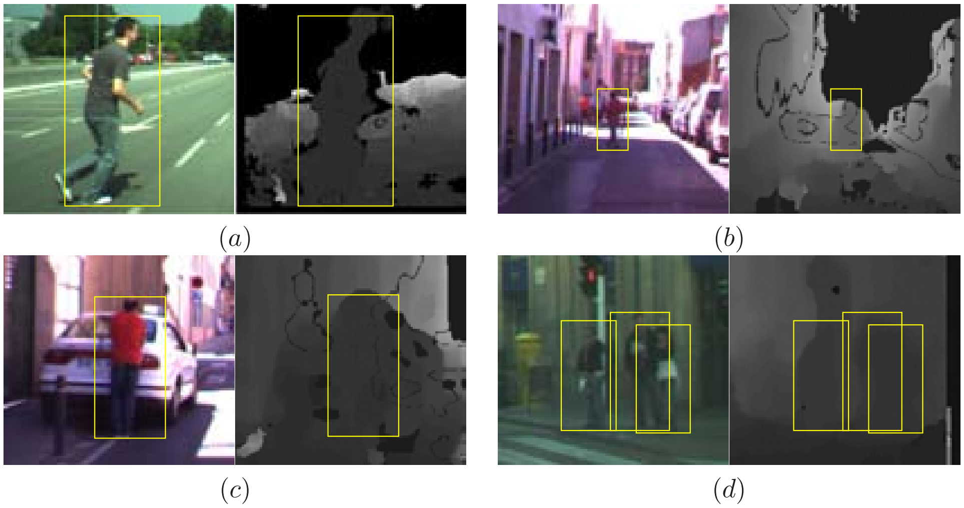

($a$)
($b$)
($c$)
($d$)

**Figure 3.10:** Typical problems when performing stereo segmentation (brightness and contrast of depth images has been modified in order to enhance the visualization). ($a$) Ideal depth image with a well-defined target shape. ($b$) Nonexistent 3D points in the head area can lead to a wrong candidate window size. ($c$) Overlapping objects, for example, cars and pedestrians, make it hard to accurately define the candidates. ($d$) Buildings and other street furniture mix with pedestrians so it is difficult to segment them.

There exist some papers that claim the use of stereo-based segmentation as stand-alone method to perform the candidate generation [174, 144, 125], but not many of them give details regarding the algorithm employed. One of the most promising examples is [125], in which a clustering method is used to identify potential obstacles in 3D, and then candidates are placed over their image projection in 2D. In order to cope with the inaccuracy of the 3D clusters, the authors use an algorithm that places 15 different sized windows over the image projection of the cluster. This algorithm has two main drawbacks. First, although the proposed technique has proven to be robust up to 25m according to the authors, at further distances stereo precision decreases and the number of 3D points available is low, so pedestrians in the range from 25 to 50 m have a high chance of not being selected although further stages could deal with them (e.g., classification). Second, the examples in the paper correspond to low-cluttered scenarios with apparently well-defined pedestrian stereo like in (Fig. 3.10($a$)), so it is unclear how the system behaves under heavy clutter and whether the generation algorithm can tackle the typical stereo problems or not (like in Fig. 3.10($b, c, d$)). According to these problems, in this thesis we do not consider stand-alone stereo techniques as a reliable algorithm to extract candidates, so it will not be evaluated.

### 3.3.4 Adaptive road scanning

Although it has been suggested that the flat world assumption algorithm has satisfactory performance in vehicle detection in high speed ways, it is not the case for pedestrian detection in urban streets. Even if the camera is always calibrated at a constant position, we can see in Fig. 3.11 that these parameters cannot be assumed constant due to the variable road geometry and the vehicle dynamics (braking, accelerating, etc.). Even using tricks like exploring three hypothetical $\theta$, which has been successfully tested for vehicles in highway scenarios [129], the results do not improve for pedestrian detection. Besides, we have seen that stereo segmentation holds many problems to be applied as an stand-alone algorithm.

In order to overcome such limitations, we present a new algorithm that adjusts the scanning grid to a dynamic road surface.

The algorithm exploits the road plane estimation technique proposed by A.D. Sappa in [134], which uses 3D points provided by a stereo camera to estimate the road surface representing it as a plane. In fact, as we will see later, this dynamic plane estimation is equivalent to updating $\theta$ and $\phi$ at each frame. Then, we present three different methods to scan the estimated road plane, aimed at optimizing together the number of generated candidates and the potential selected pedestrians. Figure 3.12 provides an scheme of the proposed technique.

#### Algorithm

The algorithm consists of two stages: road surface estimation and road/image scanning. Algorithm 3 details how it works.

The main targets of road surface estimation are two-fold: first, to fit a surface (a plane in the current implementation) to the road; second, to compute the relative position and orientation (pose) of the camera[3] with respect to such a plane. From the relative camera-road parameters ($\phi$, $\theta$ and $h$), in most of the situations the value

[3] Also referred to as camera extrinsic parameters.

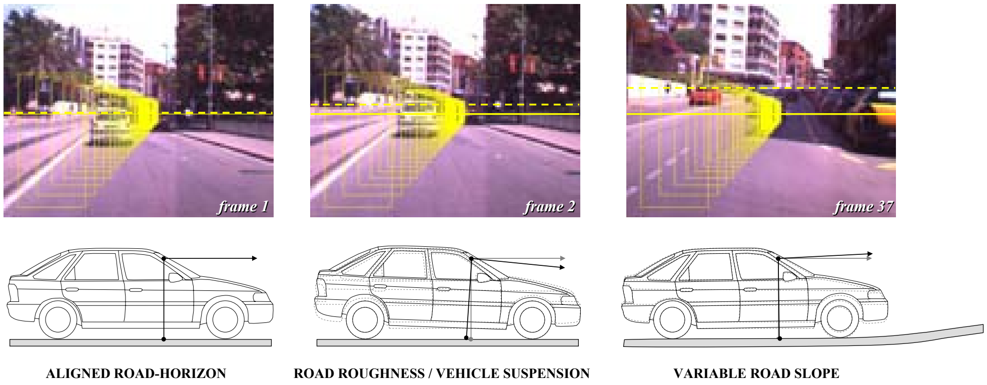


**Figure 3.11:** The incorrect aligning between the candidates and the road can be appreciated by the difference between the employed horizon line (dashed yellow) and the correct horizon line (solid yellow). The correct horizon line can be found by intersecting two parallel lines of the seen ground plane, which corresponds to the vanishing line. As can be seen, even for a single short sequence the variations in the relative camera-road positions are significant.

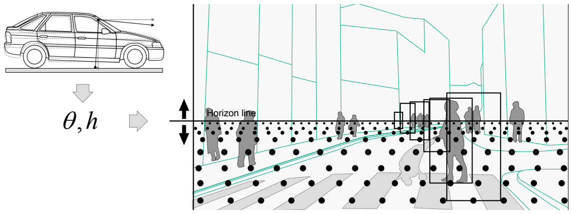


**Figure 3.12:** Scheme of the adaptive road scanning.

of $\phi$ (roll) is very close to zero. This condition is easily fulfilled by fixing a sensible $\phi$ when mounting the camera, and because in normal urban driving situations this value scarcely varies [88]. Hence, the algorithm consists in estimating $\theta$ and $h$.

We have divided this first stage in the two substages detailed below: $a$) 3D data points projection and cell selection and $b$) road plane fitting and ROIs setting[4]:

a. 3D data points projection and cell selection. Let $D(r, c)$ be a depth map provided by the stereo pair with $R$ rows and $C$ columns, in which each array element $(r, c)$ is a scalar that represents a scene point of coordinates $(x_C, y_C, z_C)$, referred to the camera coordinate system $C$ (Fig. 3.7). The aim at this first stage is to find a compact subset of points, $\zeta$, containing most of the road points. To speed up the whole algorithm, most of the processing at this stage is performed over a 2D space. Initially, 3D data points are mapped onto cells in the $(Y_C Z_C)$ plane, resulting in a 2D discrete representation $\psi(o, q)$; where $o = \lfloor D_Y(r, c) \cdot \varsigma \rfloor$ and $q = \lfloor D_Z(r, c) \cdot \varsigma \rfloor$, $\varsigma$ representing a scale factor that controls the size of the bins

[4] Please refer to [134] for more details on this stage of the algorithm.

**Algorithm 3** Adaptive Road Scanning

```
Input:
  $Z_{W_{min}}, Z_{W_{max}} \leftarrow$ closest and furthest road points in the Z axis
  $Z_N \leftarrow$ Number of scanning positions in Z axis
  $X_{W_{min}}, X_{W_{max}} \leftarrow$ scanning range in X axis
  $\Delta_x \leftarrow$ scanning stride in X axis
  $W_N$ different window sizes for each scanning point
Algorithm:
  Road Surface Estimation
    3D data point projection and cell selection
    Road plane fitting
      Dominant 2D Straight Line Parametrisation
      Road Plane Parametrisation
  Road/Image Scanning
```

according to the current depth map (Fig. 3.13). The scaling factor is aimed at reducing the projection dimensions with respect to the whole 3D data in order to both speed up the plane fitting algorithm and be robust to noise. It is defined as: $\varsigma = ((R + C)/2)/(\Delta X + \Delta Y + \Delta Z)/3)$; $(\Delta X, \Delta Y, \Delta Z)$ is the working range in 3D space. Every cell of $\psi(o, q)$ keeps a reference to the original 3D data points projected onto that position, as well as a counter with the number of mapped points.

From that 2D representation, one cell per column (i.e., in the Y-axis) is selected, relying on the assumption that the road surface is the predominant geometry in the given scene. Hence, it picks the cell with the largest number of points in each column of the 2D projection. Finally, every selected cell is represented by the 2D barycenter $(0, (\sum_{i=0}^{n} y_{C_i})/n, (\sum_{i=0}^{n} z_{C_i})/n)$ of its $n$ mapped points. The set of these barycenters defines a compact representation of the selected subset of points, $\zeta$. Using both one single point per selected cell and a 2D representation, a considerable reduction in the CPU time is reached during the road plane fitting stage.

b. Road plane fitting. The outcome of the previous substage is a compact subset of points, $\zeta$, most of them belong to the road. As stated in the previous subsection, $\phi$ (roll) is assumed to be zero, hence the projection is expected to contain a dominant 2D line corresponding to the road together with noise coming from the objects in the scene.

As a first step, the dominant line corresponding to the road is parametrized by a 2D straight line. It uses a RANSAC based [52] fitting applied over 2D barycenters intended for removing outlier cells. Initially, every selected cell is associated with a value that takes into account the amount of points mapped onto that position. This value will be considered as a probability density function. The normalized

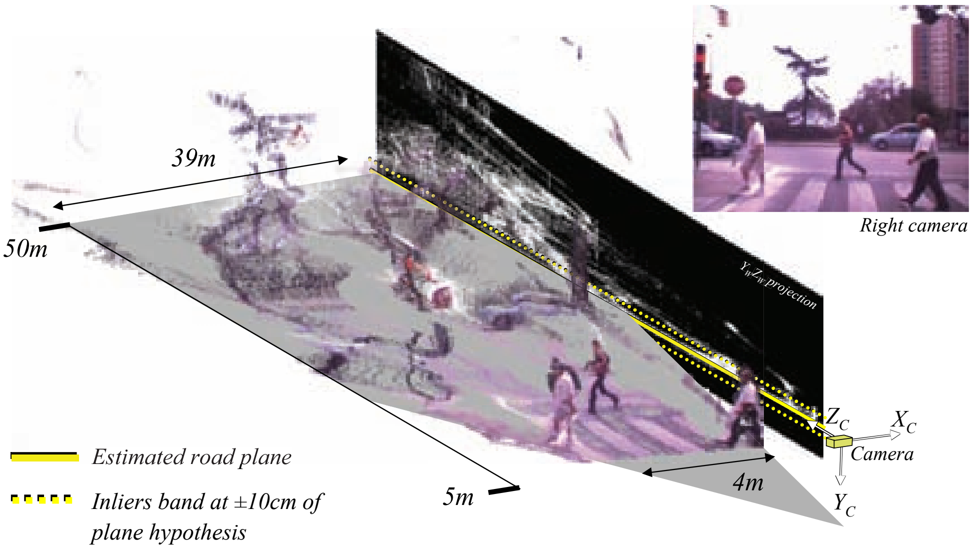


**Figure 3.13:** YZ Projection and road plane estimation.

probability density function is defined as follows: $pdf_i = n_i/N$; where $n_i$ represents the number of points mapped onto the cell $i$ and $N$ represents the total amount of points contained in the selected cells. Next, a cumulative distribution function, $F_j$, is defined as: $F_j = \sum_{i=0}^{j} pdf_i$; If the values of $F$ are randomly sampled at $n$ points, the application of the inverse function $F^{-1}$ to those points leads to a set of $n$ points that are adaptively distributed according to $pdf_i$. The algorithm is detailed in Algorithm 4.

In order to speed up the process, a predefined threshold value for inliers/outliers detection has been defined (a band of ±10 cm is enough for taking into account both data points accuracy and road planarity); an automatic threshold could be computed for inliers/outliers detection, following robust estimation of standard deviation of residual errors [132]. However, it would increase CPU time since robust estimation of standard deviation involves computationally expensive algorithms (e.g., sorting functions).

**Algorithm 4** Random Step: Straight line parametrisation

```
For $l = 1, \ldots, L$
  1. Draw a random subsample of 2 different barycenter points $(P_1, P_2)$ according
     to the probability density function $pdf_i$ using the above process;
  2. Compute the straight line parameters $(\alpha, \beta)_l$ for this subsample;
  3. For this solution, compute the number of inliers among the entire set of bary-
     center points contained in $\zeta$, as mentioned above using a ±10 cm margin.
```

The second step of the fitting is aimed at refining the plane parameters by using all the 3D data points contained into inliers cells. The algorithm (Algorithm 5) uses least squares fitting, and outputs the plane $(a, b, c)$.

**Algorithm 5** Consensus Step: Road Plane Parametrisation

```
1. From the previous 2D straight line parametrisation choose the solution that has the highest number of inliers;
2. Compute $(a, b, c)$ plane parameters by using the whole set of 3D points contained in the cells considered as inliers, instead of the corresponding barycenters. To this end, the least squares fitting approach [161], which minimizes the square residual error $(1 - ax_C - by_C - cz_C)^2$ is used;
3. In case the number of inliers is smaller than 40% of the total amount of points contained in $\zeta$ (e.g., severe occlusion of the road by other vehicles), those plane parameters are discarded and the ones corresponding to the previous frame are used as the correct ones.
```

The second stage of the algorithm is road/image scanning. Once the road is estimated, the candidates can be placed on the surface using the same method as in Flat World Assumption. The parameters are the sampling ranges $(Z_{W_{min}}, Z_{W_{max}})$, $(X_{W_{min}}, X_{W_{max}})$ and the window strides $\Delta_X$ and $\Delta_Z$. In this case, however, we use a new variable $Z_N$, which defines the number of sampling points in the $Z_W$ direction, instead of defining $\Delta_Z$. Although both parameters are related, for the sake of comparison with the other methods we propose here we will use $Z_N$.

In this case we can use the estimated plane parameters $(a, b, c)$, so the projection Equations 3.4-3.3 can be simplified to:

$$\begin{aligned}
c_w &= f_x \cdot \frac{c_W}{c_Z} && (3.6)\\
c_h &= f_y \cdot \frac{c_H}{c_Z} && (3.7)\\
c_y &= x_0 + f_x \cdot \frac{c_X}{c_Z} && (3.8)\\
c_x &= y_0 + \frac{f_y}{b \cdot c_Z} - \frac{f_y{\cdot}c}{b} - \frac{c_H}{2}\ , && (3.9)
\end{aligned}$$

where $c_z = Z_{W_{min}} + i\Delta_Z \ \forall i \in \{0, .., n_z - 1\}$; and the Z stride can be computed with $\delta_Z = (Z_{Cmax} - Z_{Cmin})/N_z$. Similarly to Eq. 3.4, the $y$ coordinate also corresponds to the window center. Hereinafter, this scheme is referred to as *uniform road scanning.*

As can be appreciated in Fig. 3.14($a$), this scheme has two main drawbacks: it oversamples far positions (i.e., Z close to $Z_{W_{max}}$) and undersamples near positions (i.e., the sampling is too sparse when Z is close to the camera). Thus, it is clear that the sampling cannot only rely on the world but shall be focused on the image. In fact, the sampling is aimed at extracting candidates in the 2D image. According to this, we compute the minimum $v_{min}$ (image projection of $Z_{W_{max}}$) and maximum $v_{max}$ (image projection of $Z_{W_{min}}$) image rows corresponding to the $Z$ range:

$$h_{\text{min}} = y_0 + \frac{f_y}{b \cdot Z_{W_{max}}} - f_y \frac{c}{b} \ , \tag{3.10}$$

$$h_{\text{max}} = y_0 + \frac{f_y}{b \cdot Z_{W_{min}}} - f_y \frac{c}{b} \ , \tag{3.11}$$

and evenly place the sampling points between these two image rows using:

$$c_h = h_{\text{max}} + i\Delta_{im} \quad \forall i \in \{0..N_z - 1\} \ , \tag{3.12}$$

where $\Delta_{im} = (y_{Z_{W_{min}}} - y_{Z_{W_{max}}})/N_z$. In this case, the corresponding $c_Z$ in the plane, needed to compute the window size, is

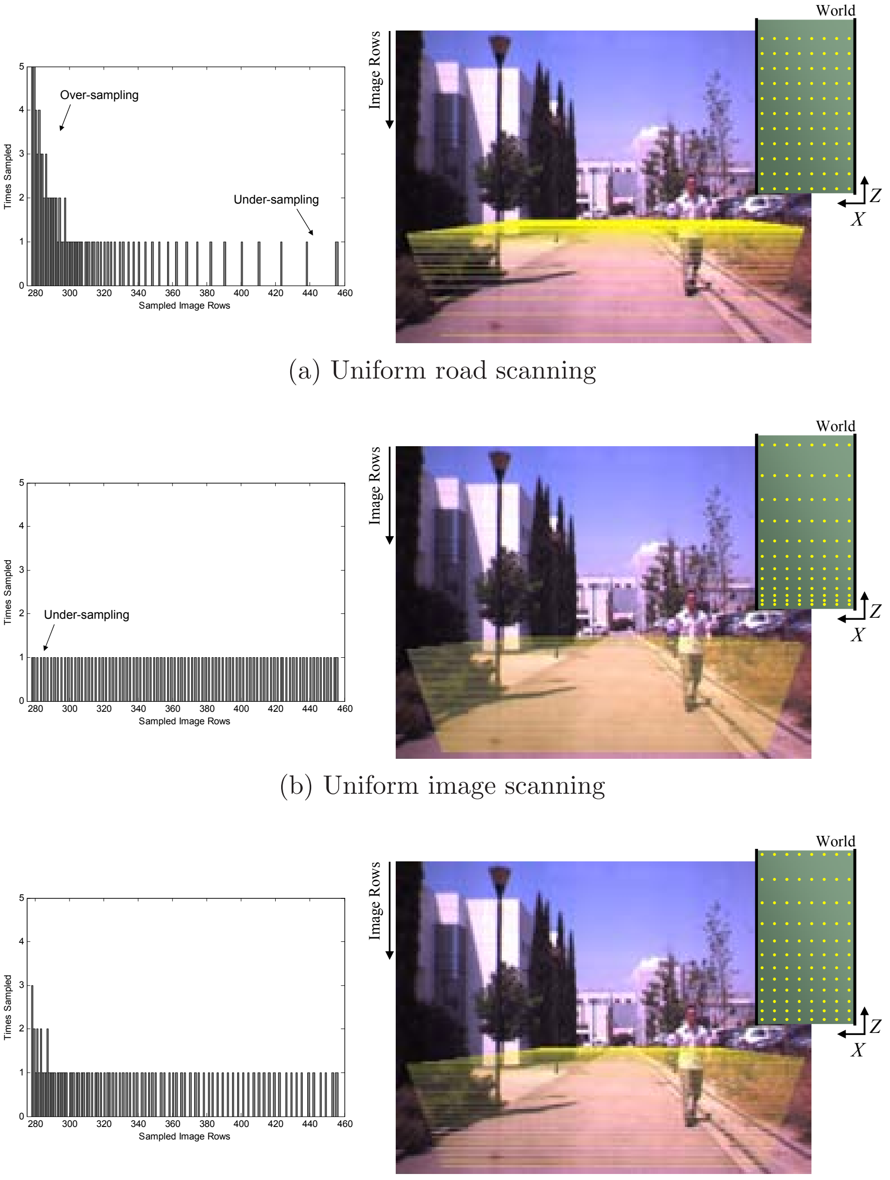
$$c_Z = \frac{f_y}{c + b(c_h - y_0)} \ . \tag{3.13}$$

In the case of $x$, $w$ and $h$, the same equations as in the first scheme can be used (3.9,3.6 and 3.7). This scheme is called *uniform image scanning*. In this case, it is seen in Fig. 3.14($b$) that although the density of sampling points for the closer $Z_W$ is appropriate, the far $Z_W$ are undersampled, i.e., the space between sampling points is too big (see histogram of the same figure).

Figure 3.15 displays the sampling functions with respect to the $Z_W$ scanning positions and the image $Y$ axis. The *uniform image scanning*, in dotted-dashed-blue, draws a linear function since the windows are evenly distributed over the available rows. On the contrary, the *uniform road scanning*, in dashed-red, takes the form of an hyperbola as a result of the perspective projection. The aforementioned over- and under-sampling in the top and bottom regions of this curve can be also seen in this figure. Attending to the problems of these two approaches, we finally propose the use of a non-uniform scheme that provides a more sensible sampling, that is, neither over- nor under-sampling the image or the world. The idea is to sample the image with a curve in between the two previous schemes, and adjust the row-sampling according to our needs, i.e., mostly linear in the bottom region of the image (close $Z$) and logarithmic-like for further regions (far $Z$), but avoiding over-sampling. In our case, we use a quadratic function of the form $y = ax^2 + bx + c$, constrained to pass through the intersection points between the linear and hyperbolic curves and by a user defined point $(i_{\text{user}}, y_{\text{user}})$ between the two original functions. The curve parameters can be found by solving the following system of equations:

$$\begin{bmatrix} {i_{max}}^2 & i_{max} & 1 \\ {i_{min}}^2 & i_{min} & 1 \\ {i_{user}}^2 & i_{user} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} v_{min} \\ v_{max} \\ y_{user} \end{bmatrix} \ , \tag{3.14}$$

where $i_{min} = 0$ and $i_{max} = N_z - 1$. For example, in the non-uniform curve in Fig. 3.15 (solid-black line), $y_{user} = i_{min} + (i_{\text{max}} - i_{min}) \times \kappa$ and $i_{user} = i_{min} + (i_{max} - i_{min}) \times \lambda$, where $\kappa = 0.6$ and $\lambda = 0.25$. For the $X_C$ axis we follow the same procedure as with the other schemes. The resulting scanning, called *non-uniform scanning*, can be seen in Fig. 3.14($c$).



(a) Uniform road scanning

(b) Uniform image scanning

(c) Non-uniform scanning

**Figure 3.14:** The three different scanning schemes. Right column shows the scanning rows using the different schemes and also a representation of the scan over the plane. In order to enhance the figure visualization just 50% of the lines are shown. The histograms of sampled image rows are shown on the left column; under- and over-sampling problems can be seen.

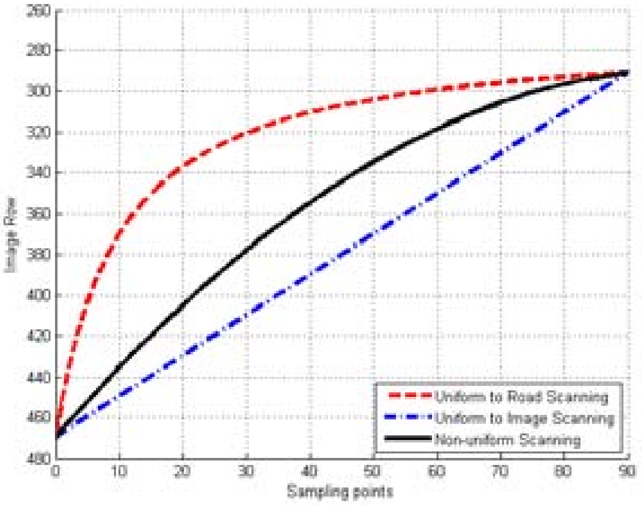


**Figure 3.15:** Scanning functions. A non-uniform scanning with parameters $\kappa = 0.6$ and $\lambda = 0.25$ is between the uniform to road and to image curves, hence achieving a more sensible scan.

### Performance evaluation

Figure 3.16 illustrates the performance of the algorithm when using each of the three different schemes and different number of sampling points in $Z_W$. The algorithm reaches a $TPR$ of nearly 0.8 with 25 000 to 75 000 Non Pedestrian Candidates. It can be appreciated that the three schemes follow a similar pattern when decreasing $Z_N$. This can be explained by the candidates that are left out of the image, which of course are not taken into account. In the case of *uniform image scanning* sampling, given that the candidates are distributed along the image instead of the road, there are more candidates closer to the camera than with the non-uniform scanning. Since the close candidates that are out of the image (because of their $W_X^C$) are directly discarded and not taken into account, the number of NPC is smaller. The same happens with non-uniform scanning with respect to *uniform road scanning* scheme. It is also worth to mention that for this experiment we use $\mathcal{S} = 5$, as with the flat world assumption algorithm. If $\mathcal{S}$ is reduced to 1 the performance decreases in about a 30% of the TPR in every case.

We have selected the non-uniform scanning with $Z_N = 50$ as the representative configuration of this algorithm (marked in gray in Fig. 3.16).

The three schemes have a satisfactory reduction in the number of candidates, between 10 000 and 50 000, when $Z_N = 25$ or 50, we select the non-uniform scanning with $Z_N = 50$, which provides the higher DR in this range and more than 4 CPA. In addition, the non uniform one provides a finer sampling for further distances than the uniform to image one, which is an important point to be considered.

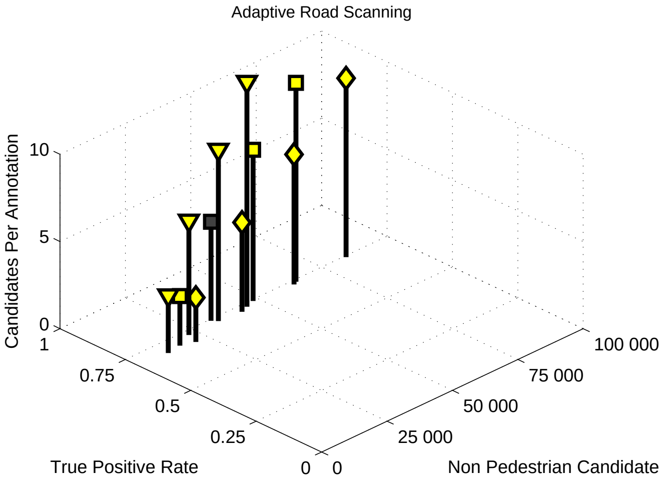


**Figure 3.16:** Adaptive Road Scanning performance plot. ◇ represent uniform road sampling, △ represent uniform image scanning and □ represent the non uniform scanning.

### 3.3.5 Adaptive road scanning with 3D-based filtering

Adaptive road scanning provides a significant performance improvement compared to the previous algorithms. However, the potential of stereo seems not to be fully exploited. In this new proposal we extend the adaptive road scanning with an extra-step that filters candidates based on stereo data. This step is aimed at discarding the candidate windows that are empty of 3D points, which in fact will not correspond to pedestrians (Fig. 3.17). The method starts by aligning the camera coordinate system with the world coordinate system (see Fig. 3.7) with the aim of compensating pitch angle $\theta$, computed in Sect. 3.3.4. Assuming that roll is set to zero, as described in the aforementioned section, the coordinates of a given point $p_{(x,y,z)}$, referred to the new coordinate system, are computed as follows:

$$\begin{aligned} p_{x_R} &= p_x \\ p_{y_R} &= cos(\theta)p_y - sin(\theta)p_z \\ p_{z_R} &= sin(\theta)p_y + cos(\theta)p_z \ \ . \end{aligned} \tag{3.15}$$

Then, rotated points located over the road[5] are projected onto a uniform grid $G_P$ in the fitted plane (Sect. 3.3.4), where each cell has a size of $\sigma \times \sigma$. A given point $p(x_R, y_R, z_R)$ votes into the cell $(i, j)$, where $i = \lfloor x_R/\sigma \rfloor$ and $j = \lfloor z_R/\sigma \rfloor$. The resulting map $G_P$ is shown in Fig. 3.18($b$). As can be seen, cells far away from

[5] Set of points placed in a band from 0 to $2m$ over the road plane, assuming that this is the maximum height of a pedestrian.

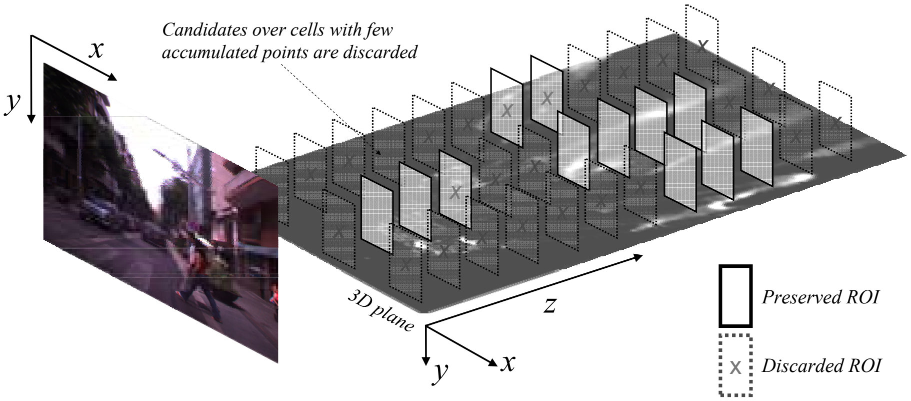

Candidates over cells with few accumulated points are discarded
x
y
3D plane
z
y
x
Preserved ROI
x
Discarded ROI

**Figure 3.17:** Schematic illustration of the candidate filtering stage.

the sensor tend to have few projected points. This is caused by two factors. First, the number of projected points decreases directly with the distance, as a result of perspective projection. Second, the uncertainty of stereo reconstruction also increases with distance, thus the points of an ideal vertical and planar object would spread wider into $G_P$ as the distance of these points increases. In order to amend these problems, the number of points projected onto each cell in $G_P$ are reweighted and redistributed. The reweighting function is

$$G_{RW}(i,j) = j\sigma G_P(i,j) \ , \tag{3.16}$$

where $j\sigma$ corresponds to the real depth of the cell. The redistribution function consists in propagating the value of $G_{RW}$ to its neighbors as follows:

$$G(i,j) = \sum_{s=i-\eta/2}^{i+\eta/2} \sum_{t=j-\eta/2}^{j+\eta/2} G_{RW}(s,t) \ , \tag{3.17}$$

where $\eta$ is the stereo uncertainty at a given depth (in cells): $\eta = \text{uncertainty}/\sigma$. Uncertainty is computed as a function of disparity values:

$$\text{uncertainty} = f_y \cdot \text{baseline} \frac{\mu}{\text{disparity}^2} \ , \tag{3.18}$$

where baseline is the one of the stereo rig in meters and $\mu$ is the correlation accuracy of the stereo. The resulting map $G$, after reweighting and redistribution processes, is illustrated in Fig. 3.18($c$). The filtering consists in discarding the candidate windows that are over cells with less than $\chi$ points, which is set experimentally. In our implementation, this parameter is low in order to fulfill the conservative criterion mentioned in the introduction, that is, in this early stage potential false positives are preferred than immediate false negatives.

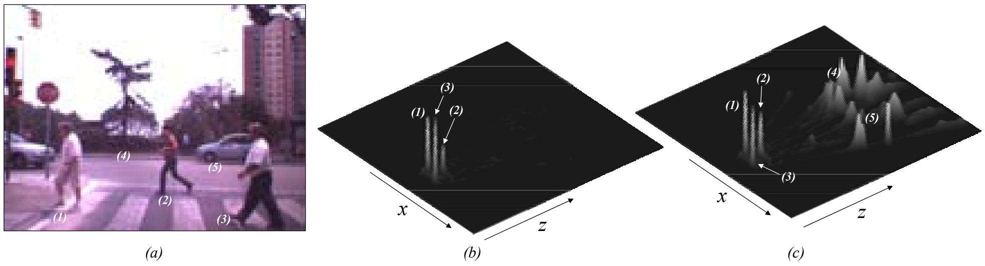


**Figure 3.18:** Probability map of vertical objects on the road plane. ($a$) Original frame. ($b$) Raw projection $G_P$. ($c$) Reweighted and redistributed vertical projection map of the 3D points of the frame.

### Performance evaluation

In order to test the performance of the filtering, we have used the original adaptive road scanning with the non-uniform scanning scheme and $Z_N = 50$. Figure 3.19 illustrates the significant reduction in NPC with respect to the original algorithm, i.e., the number of candidates is reduced from 34 000 without filtering to around 10 000 with filtering (using a very low threshold), at the price of only reducing a 8% the TPR.

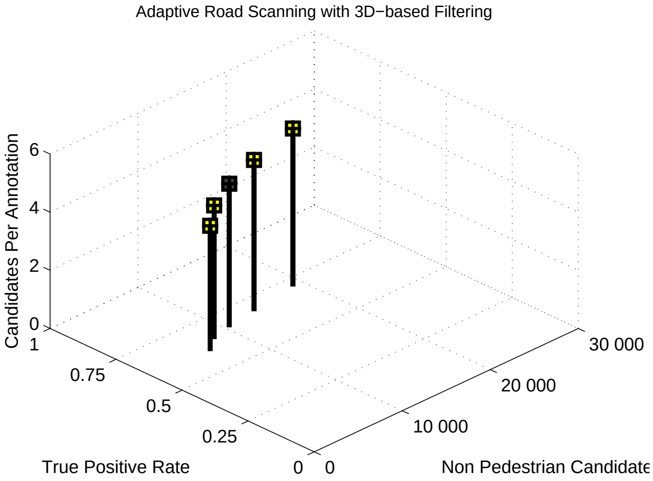


**Figure 3.19:** Adaptive road scanning with 3D-based filtering performance plot.

### 3.3.6 Probabilistic 3D scanning

Regardless of the good results of adaptive road scanning both with and without the 3D-based filtering, their TPR is still not as high as would be desirable, specially for far pedestrians. One of the reasons is that in some cases the estimated plane is not perfectly adjusted to the actual road, for example, the road is not a plane but a more complex surface. Furthermore, as can be seen in Fig. 3.19, the stereo filtering also implies a small reduction of the TPR. In this case, the furthest pedestrians are again the principal source of FN, presumably as a result of the aforementioned lack of precission of stereo for far distances. In addition, it shall be stated clear that the performance of these algorithms cannot be improved by relaxing parameters, that is, the stated performance represents is the optimum one from the whole spectrum of parameter configurations.

In our last proposal we present a probabilistic model that aims at tackling the aforementioned lacks. The idea is to build a model that takes into account both the plane and the stereo uncertainty according to distance. Contrary to the adaptive road scanning, in this algorithm we are not restricting the scan to the estimated plane but expand it to all the three dimensions of the scene, and use both the plane and 3D points observations to filter out candidates. Figure 3.20 illustrates the proposal.

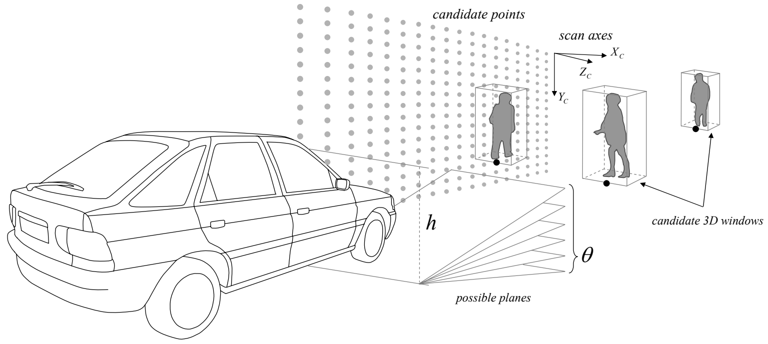


**Figure 3.20:** Scheme of the probabilistic 3D scanning algorithm.

#### Algorithm

The algorithm starts with a 3D scanning in all X-Y-Z dimensions, which provides a set of 3D precandidate positions. These positions are fixed for all the frames since they are not constrained to the road geometry. They form a grid in front of the vehicle, which in our case is uniform in X and Y axes, and non-uniform in the Z one (we use the sampling scheme of Sect. 3.3.4). Each precandidate has world dimensions of $1.70 \times 0.85 \times 1$ m, and if it is not discarded by the probabilistic model, it generates $\mathcal{S}$ different candidates, as in the previous algorithms.

The proposed probabilistic model consists of three observations. The first one is the distance from the camera to the precandidate, $z$, which is set by the 3D scanning. The second one corresponds to the amount of 3D points contained in the precandidate,

$v$ (from *voxels*), which will be fast computed by an integral volume algorithm that is described later. The third one is not related to a precandidate but to the road geometry, and corresponds to the number of points that support a set of hypothetic planes, each defined by a pitch $\theta$. These observations are then fed to a Bayesian Network that computes the probability of a precandidate to correspond to a 3D object located on the road plane, i.e., a candidate. The formalization of the method is detailed in Algorithm 6.

**Algorithm 6** Probabilistic 3D Scanning

```
Input:
  $Z_{W_{min}}, Z_{W_{max}} \leftarrow$ closest and furthest road points in the Z axis
  $Z_N \leftarrow$ Number of scanning positions in Z axis
  $X_{W_{min}}, X_{W_{max}} \leftarrow$ scanning range in X axis
  $\Delta_x \leftarrow$ scanning stride in X axis
  $Y_{W_{min}}, Y_{W_{max}} \leftarrow$ scanning range in Y axis
  $\Delta_y \leftarrow$ scanning stride in Y axis
  $\mathcal{S}$ different window sizes for each scanning point
  $\theta_{min}, \theta_{max}, \Delta_\theta \leftarrow$ range and stride of hypothesized planes pitch angles
  $\eta_{Z_{min}}, \eta_{Z_{max}} \leftarrow$ parameters of the distance to plane function (Fig. 3.24($left$))
  $v_{Z_{min}}, v_{Z_{max}} \leftarrow$ parameters of the number of 3D points function (Fig. 3.24($right$))
Algorithm:
  Compute Integral Volume using the 3D points
  Use algorithm of Sect. 3.3.4 to estimate $\theta$ and $h$
  Compute a set of planes $\mathcal{P}$ around the estimated $\theta$ and compute their support $\xi$
  $\mathcal{W}^c \leftarrow$ Generate the precandidates in the 3D scene
  Fill with the available observations from $\mathcal{P}$ and $\mathcal{W}^c$
  Compute the state of the precandidates and filter them accordingly
```

### Plane parameters

The algorithm makes use of the road plane estimation method described in Sect. 3.3.4, which provides the estimated plane parameters $(a, b, c)$. As was previously mentioned, these parameters correspond to the relative camera-road pitch $\theta$ and height $h$. In order to deal with the uncertainty of the road estimation, in the current algorithm we also take into account a set of slightly different planes and compute their supporting inliers. Figure 3.21 illustrates the support that a set of planes around the estimated have. If a $h$ is fixed to its estimated value (Fig. 3.21($c$)), then the ordinates axis draws a function $\xi$ that indicates the number of inliers out of the total number of projected cells $\zeta$ that is contained in the given plane by $\theta$. Given that the function draws a nearly constant shape independently of $h$ (see ($b$)), we will assume the estimated $h$ to be correct, so it will not be necessary to include it in the probabilistic model.

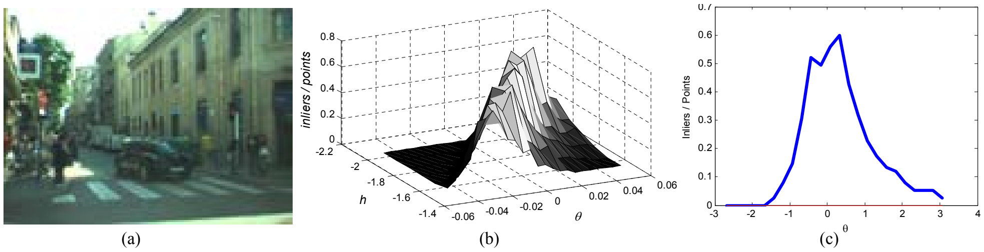


**Figure 3.21:** (*a*) Test frame. (*b*) Support for a set of planes when varying both $\theta$ and $h$. (*c*) Support for a set of planes when varying $\theta$.

### Integral volume representation

One of the observations of the proposed model is the amount of 3D points contained in a precandidate, defined as $v$. In order to compute this value it would be necessary to check if each one of the 3D points is inside the window. If we assume the input image to be $640 \times 480$, and that each pixel contains 3D information, we should check more than 300 000 points per window. This represents a heavy bottleneck for our application. In order to fast compute $v$ we use the integral volume ($iv$) representation [84]. This representation is similar to the so-called integral image ($ii$) representation widely used to compute rectangular features, as for instance Haar features, using just a few memory accesses. In this case, the $iv$ is computed in the same way that $ii$ but with 3 dimensions instead of 2, which adds some extra-operations.

The $iv$ representation is used as follows. First, voxels are initialized at 0. Then, each 3D point is accumulated in the voxels. The $iv$ transform is then computed by scanning the input volume and using the following equation:

$$iv_8 = (v_8 + iv_4 + iv_6 + iv_7 + iv_1) - (iv_2 + iv_3 + iv_5) \ , \tag{3.19}$$

where $v$ represents the original voxel. Figure 3.22($a$) illustrates the relative position of the voxels involved in this equation. Finally, the sum of the voxels inside a given volume is computed with the equation

$$(iv_8 + iv_2 + iv_3 + iv_5) - (iv_4 + iv_6 + iv_7) \ . \tag{3.20}$$

### Belief propagation

Figure 3.23 illustrates the proposed Bayesian Network that models the variables, observations and dependencies implied in the problem. The observations are represented as gray circles, while hidden (also called latent) variables are white. The directed arrows express causality between two variables and represent the conditional probabilities [22]. The left side of the network represents the plane variables and consists of an observation $\xi$, which is the proportion of inliers that supports each given plane with respect to all the points in $\zeta$; and hidden variable $\theta$, which is the pitch angle of the given plane. The right part of the network represents all the possible $N$ candidates, so it is repeated for all the windows. Each one consists of a known $z$

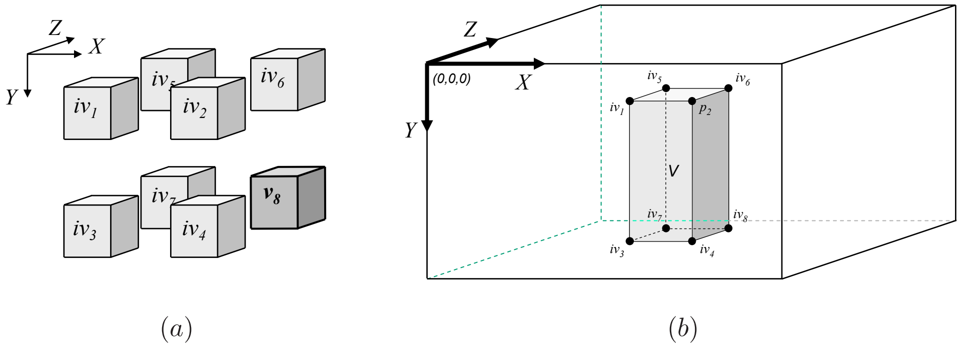


**Figure 3.22:** $(a)$ Computation of the $iv$. The darker box represents the position of $iv_8$ voxel. $(b)$ Retrieval of the sum of a volume.

distance from the camera and $v$ points in its 3D volume, an unknown $\eta$ distance to the plane, and $s$, which is the final binary value that defines whether the precandidate has to be selected as candidate or not.

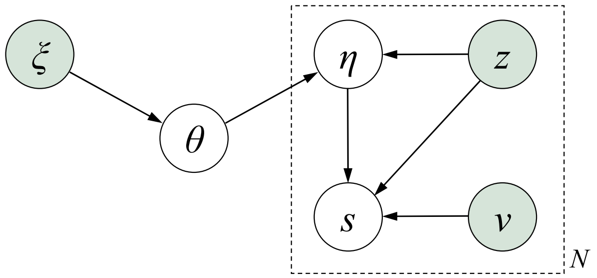


**Figure 3.23:** Bayesian Network used for Algorithm 6. See text for details.

Each edge of the network defines a conditional probability that relates two variables:

- $p(\theta|\xi)$ is the probability of a plane (defined by pitch $\theta$) given the inliers supporting it $\xi$.
- $p(\eta|\theta, z)$ is the probability of $\eta$ to be the distance from the precandidate to the plane, given $\theta$ and the camera-precandidate distance $z$.
- $p(s|\eta, z, v)$ is the probability that a given precandidate is selected (not filtered) given its distance to the plane $\eta$, its distance to the camera $z$, and the number $v$ of 3D points inside its volume. We have modeled the probabilities $p(s|\eta, z)$ and $p(s|v, z)$ as the linear functions showed in Fig. 3.24. In both cases the functions are set experimentally. In the first case, the probability of choosing a precandidate given its distance to the plane rises with $z$, since the plane estimation error grows in the same way. In the second case, the amount of 3D

points required for a close precandidate is higher than with further ones in order to compensate perspective projection, similarly to Sect. 3.3.5. It can be seen as a threshold that is decreased with $z$.

- $p(\theta)$ is the prior probability that a given plane is the correct one. We assume a normal distribution in which $\mu_\theta = \frac{\theta_{max}-\theta_{min}}{2}$, $\sigma_\theta =$, where $\theta_{max} = 5°$ and $\theta_{min} = -5°$ are the upper and lower limits of the pitch angle.

- $p(s)$ is the prior probability of a precandidate to be selected. We assume $p(s) = 0.4$, however, for future research it is desirable to make an study to analyze how many of the precandidates tend to be finally selected.

Finally, we make use of the Max-Sum algorithm [22], defined as

$$\arg\max_{s,\theta,h} \; p(s,\theta,h|\xi,z,v) \;\; , \tag{3.21}$$

which computes the most probable states of $s$, $\theta$ and $\eta$ given the observations.

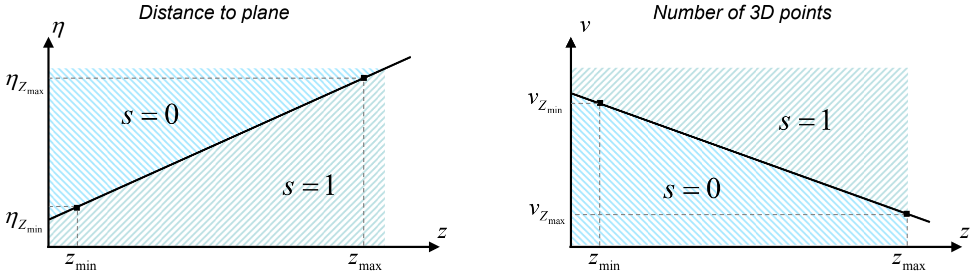


**Figure 3.24:** Functions used to define $p(s|\eta, z)$ and $p(s|v, z)$ according to the distance and number of 3D points contained in the precandidate volume.

### Performance evaluation

For the experiments we make use of a 3D grid defined by the following parameters: $Z_{W_{min}} = 0$ m, $Z_{W_{max}} = 50$ m, $X_{W_{min}} = -10$ m, $X_{W_{max}} = 10$ m, $Y_{W_{min}} = -1$ m, $Y_{W_{max}} = 5$ m, $Z_N = 25$, $\Delta_x = 0.1$ m and $\Delta_y = 0.2$ m. For the candidates placement in the $Z$ axis we use the non-uniform scheme presented in section Sect. 3.3.4. Each one of the windows in the grid is evaluated by an inference software developed by Dr. J Serrat. It is a generic non-optimized software to be used for any type of graphical model and any configuration. After all the windows are evaluated, we generate just one candidate from each, so $W_N = 1$.

Figure 3.25 illustrates the performance of the algorithm. As can be seen, it outperforms all the previous techniques: the TPR is close to sliding window but the number of NPC is similar to the adaptive road scanning. In fact, while the number of NPC is around 30 000 like the latter, the TPR is a 30% higher.

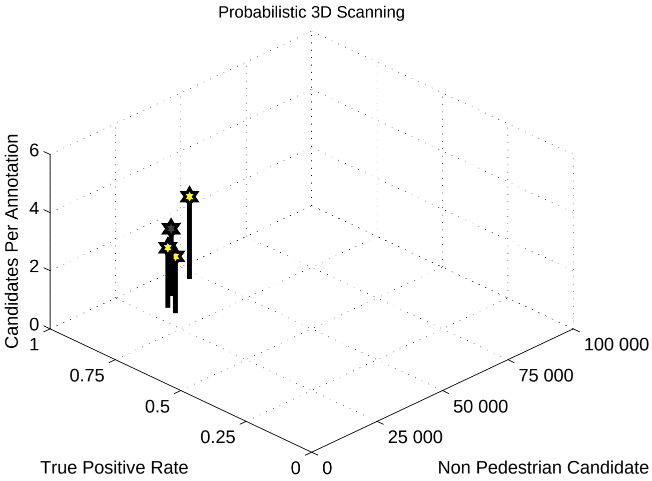


**Figure 3.25:** Probabilistic 3D Scanning performance plot.

### 3.3.7 *v*-disparity-based approaches

During the last years, the *v*-disparity representation, proposed by Labayrade et al. [89], has gained relevance in the ADAS research area. It consists in transforming the disparity image into another space in which an hypothetic road plane becomes a straight diagonal line and vertical obstacles in the road become straight vertical lines (Fig. 3.26). This method has been successfully tested for vehicle detection in relatively flat highways [89]. Later on, Broggi et al. propose to use the *v*-disparity for pedestrian detection in unstructured environments. In their case, the pedestrians are localized by the use of histograms of the *v*-disparity image and clustering techniques.

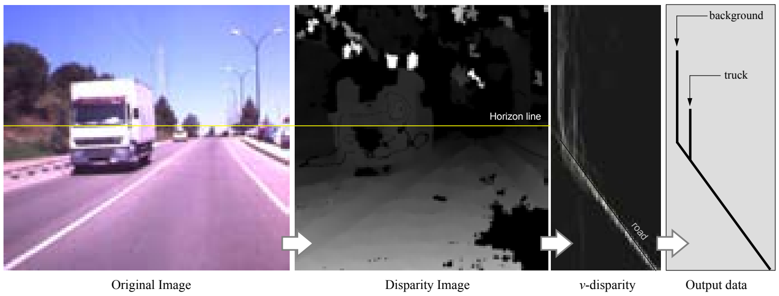


**Figure 3.26:** *v*-disparity computation. First, the disparity image pixels are row-wise accumulated in a new space, where the X axis represents disparity and Y is the original pixel ordinate. Hough transform can be used to extract the road slope (yellow), which indicates the horizon line Y position. Both the background (disparity= 0) and the obstacles are represented by vertical lines which can be also detected by the use of Hough Transform.

In the aforementioned papers, the presented results show the algorithm working in highways and open-areas with few objects in the scene (similar to Fig. 3.27(a)). We have tested the algorithm performance in our dataset, consisting of cluttered urban-scenes. To the best of our knowledge, at this moment there is no literature testing the algorithm under these conditions, so we point out some important aspects:

- First, we have noted that the computed road slope is very similar to the one computed with the proposed road surface estimation algorithm of Sect. 3.3.4. Moreover, if we use the estimated road slope and the proposed scanning schemes, we get a similar candidate generation performance plot to the adaptive road scanning. In fact, it can be said that the two methods have their advantages and disadvantages. For instance, in [135] it can be seen that in non-planar surfaces the error tends to be higher in $v$-disparity. In the case of road surface estimation, it is computationally slower and requires stereo reconstruction. However, one has to keep in mind that the 3D points are also useful for other ADAS applications, so the computation will be shared between applications.
- Second, as can be seen in Fig. 3.27$(b-f)$, the $v$-disparity is very noisy in urban scenes, so it is difficult to extract reliable windows directly from the transformed image, as could be made in non-urban scenes. In fact, large objects like vehicles have a large surface where disparities are accumulated, but in the case of medium distance pedestrians (20 m) the number of pixels decreases, which translates in few accumulated pixels in the $v$-disparity image.

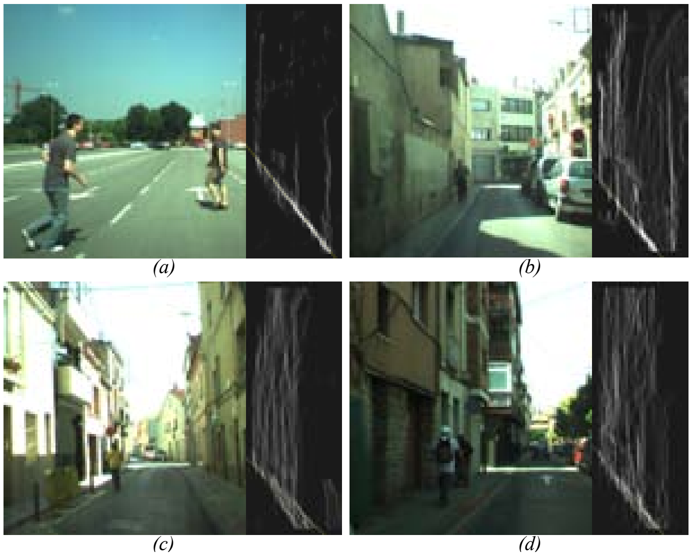


**Figure 3.27:** $v$-disparity representation of different frames. $(a)$ Scene containing isolated pedestrians and low clutter where the histogram gives clear vertical lines. $(b-d)$ In urban scenarios the histogram is too noisy so it seems difficult to be used for pedestrian detection. (The images have been cropped to enhance the visualization)

## 3.4 Experimental results

This section presents a global comparison, both quantitative and qualitative, of the algorithms proposed in this chapter for foreground segmentation. In addition, a study on their computation time is also carried out.

### 3.4.1 Quantitative evaluation

Figure 3.28 illustrates the performance comparison between the different algorithms by making use of the best parameter configuration of each one, i.e., the shaded one of each algorithm plot. From these experiments, some evidences can be highlighted:

- The reduction of NPC of all the methods with respect to the sliding window is drastic. In general using any of the proposed algorithms leads to a reduction of more than 90% candidates, hence also of NPC.
- Flat World Assumption is useless in realistic environments where the relative camera-road angle varies.
- In some cases, the algorithm will have to be chosen according to the system requirements. For instance, the candidates number reduction from 30 000 to 10 000 when using 3D-based filtering will likely translate into a significant computation time reduction, however it is worthwhile always that the system is able to accept at least a 8% of reduction in detection rate.
- Probabilistic 3D scanning stands as the most promising algorithm, it has the capacity of reducing in more than 90% the sliding window while maintatining a very high TPR.
- Regarding the distribution of candidates with respect to distances, sliding window algorithm dedicates around a 90% to the far pedestrians (from 25 to 50 m) and just a 10% for the near ones. In the proposed algorithms the effort is more balanced, for example, in the adaptive road scanning a 40% are near candidates and a 60% corresponds to far ones. From our opinion, this latter distribution is preferrable, specially if one takes into account that closer pedestrians are the most vulnerable ones and require special efforts.

### 3.4.2 Qualitative evaluation

Figure 3.29 presents illustrative results of the different algorithms evaluated in this chapter. In most of the figures the displayed candidates have been filtered to let the reader make an idea on how the algorithms work. It can be seen that the Sliding Window represents the naïve approach, i.e., it scans the frame in a blind fashion, and how the Flat World Assumption tries to reduce the number of candidates (at the price of losing pedestrians because the fixed plane parameters are incorrect). Adaptive road scanning provides a fine plane adaptation, and the 3D filtering step provides the most optimized results, i.e., no pedestrians seem to be lost and the total number of candidates is significantly reduced. The last image, representing probabilistic 3D scanning,

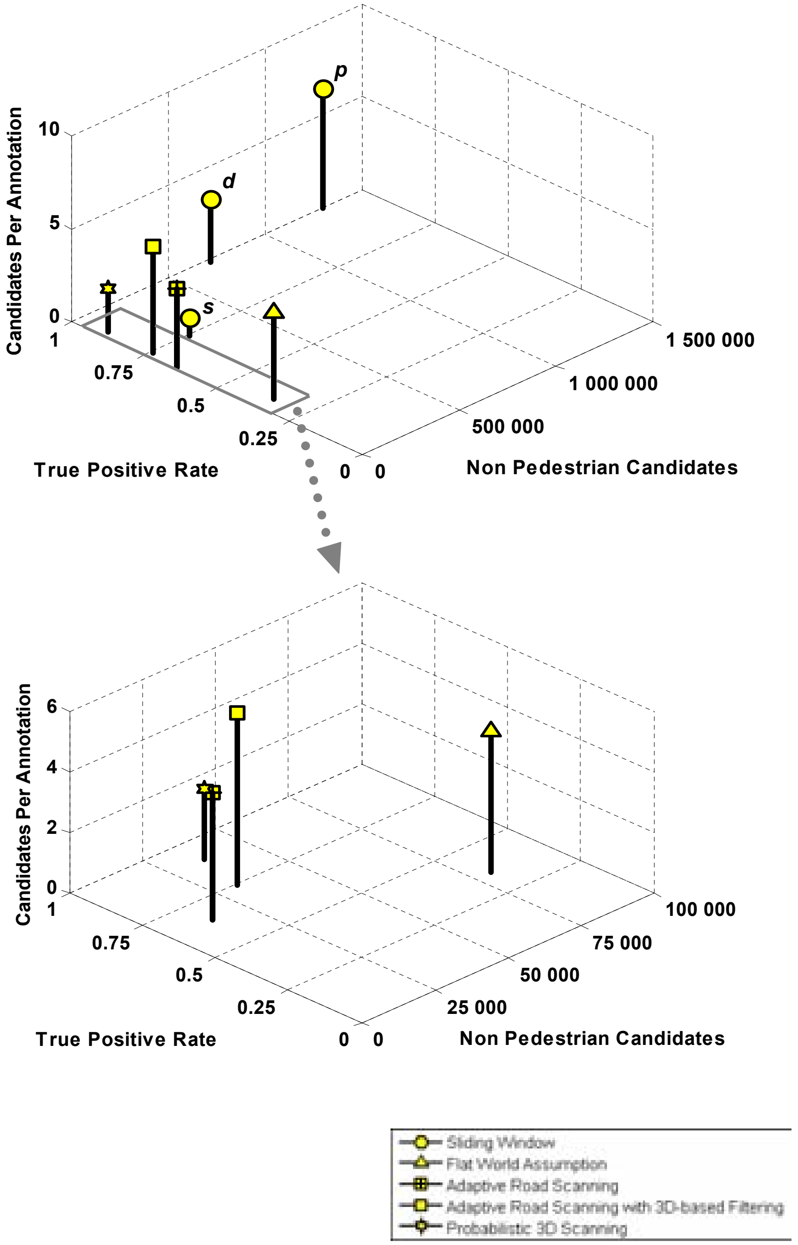
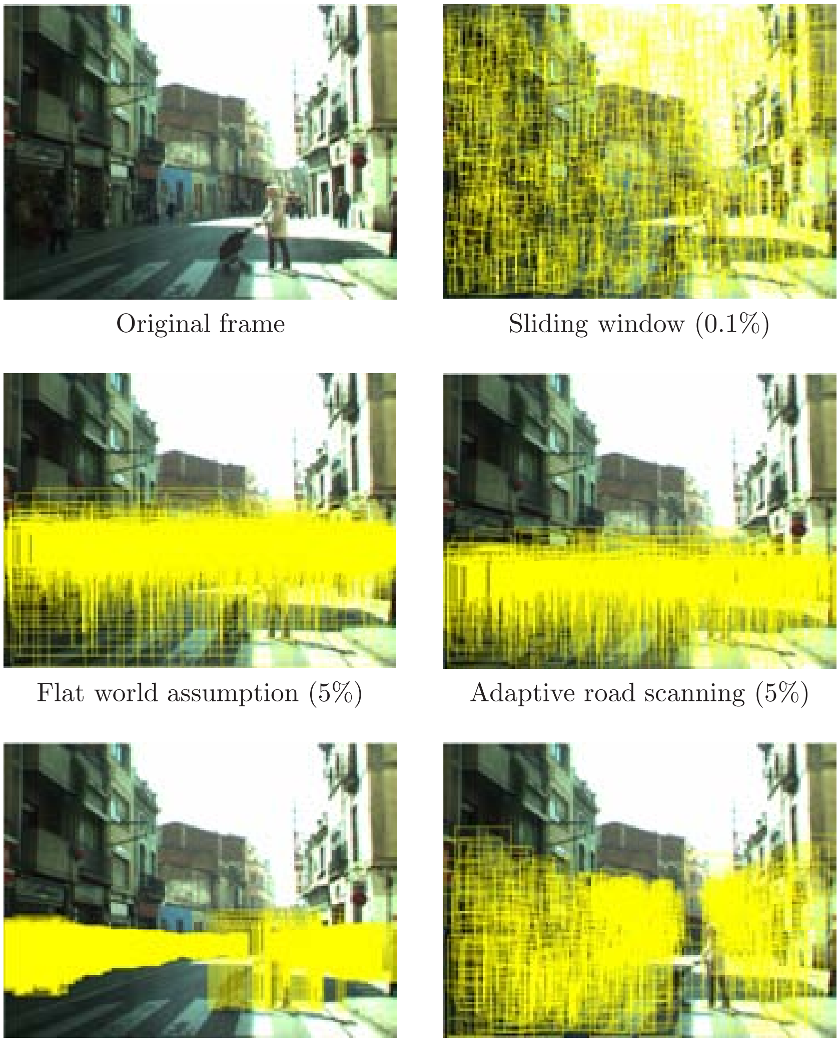


**Figure 3.28:** Algorithms performance comparison.

shows more sparse candidates than the two previous algorithms, i.e., candidates are not attached to the road, but takes into account their 3D content, which makes some areas to be ommited from the scan.



Original frame

Sliding window (0.1%)

Flat world assumption (5%)

Adaptive road scanning (5%)

ARS with 3D-based filtering (100%)

Probabilistic 3D Scanning (5%)

**Figure 3.29:** Qualitative Evaluation of the Algorithms. The number in parenthesis indicates the percentage of displayed windows.

### 3.4.3 Computation time

The final evaluation corresponds to the computation time required for each algorithm. As was previously mentioned (Chap. 2), the computation time shall not be taken as a single key to chose one or another, as optimized algorithms and new technology are always likely to reduce the timings. Again, the parameters for each algorithm correspond to the best parameters selected in each single performance analysis.

Table 3.2 details the computation time spent by each algorithm per frame. All the algorithms have been coded in C++ without any kind of special optimizations, and run in a Pentium 4 at 3.2 GHz desktop PC. In the case of Sliding Window and Flat World Assumption, the Generation Time is zero given that the candidates are the same for all the frames. In the case of the Classification Time, it can be computed by assuming different classification times for each candidate.

Adaptive Road Scanning with 3D-based filtering is the fastest algorithm, if we take into account the time required to classify the candidates. Probabilistic 3D scanning the slowest. The computation time for a frame is around 40 s, with a grid to analyze consisting of around 80 000 candidates. However, we are optimistic with respect to the possible optimizations: the current experiments make use of a generic inference software that can deal with any number of cliques, labels, etc., and the implementation makes extensive use of loops. In the current case, since the cliques are always pairwise and the number of edges, variables and labels can be defined a priori, we plan to make use of matrix computation, which would drastically accelerate the code.

In the cases in which road fitting is performed, for example, adaptive road scanning and probabilistic 3D scanning, the time spent in the by the stereo reconstruction software is around 60 ms and around 90 ms more are spent in the fitting algorithm. In the case of the 3D reconstruction, there already exist faster implementations so this time could be also reduced, and moreover, the results would not only be used by the PPSs but also by other ADAS.

**Table 3.2:** Computation time statistics of the foreground segmentation algorithms. The numbers correspond to the frames average for the CVC-02-CG dataset. The column Near is the percentage of candidates smaller than 60 pixels, which correspond to the candidates from 0 to 25 m far from the car. $T_{gen}$ is the time spent in generating the candidates in each frame. $T_{class}$ is the time spent to classify the candidates assuming that the classifier takes 0.1 ms per window.

| Algorithm | #Cand. | TPR | Near | $T_{gen}$ | $T_{class}$ |
|---|---|---|---|---|---|
| Sliding Window (perfect) | 1 300 000 | 100% | 12.8% | 0 s | 130 s |
| Sliding Window (dense) | 700 000 | 98% | 12.8% | 0 s | 70 s |
| Sliding Window (sparse) | 220 000 | 75% | 9% | 0 s | 22 s |
| Flat World Assumption | 42 000 | 35% | 25% | 0 s | 4.2 s |
| Adaptive Road Scanning | 32 000 | 74% | 41% | 0.16 s | 3.2 s |
| Adapt.+ 3D-based Filtering | 7 000 | 66% | $50-70\%$ | 0.19 s | 0.7 s |
| Probabilistic 3D Scanning | 36 000 | 90% | $30-80\%$ | 40 s | 3.6 s |

## 3.5 Discussion

Although sliding window is by far the most used foreground segmentation algorithm in pedestrian detection literature, is has been demonstrated that it is not the most optimum approach. The number of windows to classify is big, which translates into a big time consumption in the later classification stage. It is true that we could assume some application-based restrictions even in sliding window, for instance directly discarding 1/3 of the top image, as pedestrians will not be found in such an area. However, according to our experiments the reduction is not that significant when compared to the number of candidates in the other methods..

From all the proposed algorithms, each one has its advantages and drawbacks, specially in terms of performance and computation time, so it is not easy to point to any as the overall best. For instance, adaptive road scanning drastically reduces the number of candidates at a low computation time spent, however, the TPR is not perfect, which means that some pedestrians will be lost in this early stage. On the contrary, in the case of probabilistic 3D scanning the TPR is almost perfect but the computation time, at this moment, is high. According to this, our proposed system in Chap. 6 will be based on adaptive road scanning techniques. However, our future research line is without doubt focused on the probabilistic framework, as it is the most promising technique.

There are some aspects that have not been analyzed and future lines that worth to be highlighted:

- All the proposed algorithms work in a per-frame basis. The next step is to make use of temporal analysis, given that we work with sequences of video. For instance, motion analysis, candidates tracking over time, are the key points to achieve maximum TPR at minimum NPC.
- Improved acquisition is a must if reliable candidate generation is to be made for $25 - 50$ m. At this moment, since the resolution of the sensor and the stereo reconstruction is low, the number of candidates to be generated at these distances is artificially high in order to provide a certain robustness. Stereo pairs with longer focal lengths and wider baselines are likely to significantly improve the performance.
- More complex models of the road are likely to improve the results, for example, piecewise road estimations, which have already been used for highways, will provide a more accurate road model, specially for far distances. Other ideas like distinguishing the pavement and the asphalt can be of great help when computing the positions of close candidates.

## 3.6 Summary

In this chapter we have demonstrated that there is still big room for improving pedestrian detection by looking at all the different tasks involved, not only classification. We have studied different foreground segmentation algorithms. Since the literature on this subject, and specially on its performance evaluation criteria, is almost nonexistent, we propose both a pioneer dataset and an evaluation protocol that should be useful for future research in this area. In addition, we formalize interesting techniques already present in the literature and propose new ones, which are used in the next chapters. Finally, we perform evaluation comparisons using the proposed protocol.

*Pace*
Prabhakara Jimmy Quek, oil on canvas, 1998
(Private Collection)

*Pace* is inspired in one of the typical busy business streets in Singapore. It depicts people walking and crossing roads in a traffic junction, which in fact is highly related to the research in this thesis. From our point of view, the painting illustrates many of the components that make the task of learning human appearance a hard not solved problem: different clothes, colors, illuminations, occlusions, people carrying objects, in addition to the own variability of people. Finally, from a psychologically viewpoint, the notion of foreground segmentation is strongly represented in the painting. As the reader can appreciate, there is a big free space that covers almost 1/4 of the image, which receives few attention in comparison to the *relevant* actors in the scene. This is related to the psychological viewpoint of human perception and to the notion of foreground segmentation, whose aim is to focus the analysis (attention) also on the relevant actors omitting the free space.

# Chapter 4

# Object classification

Classification has traditionally been the most explored task in pedestrian detection. As previously seen in Chap. 2, the literature focusing classification of pedestrians has significantly grown in the last years. However, in spite of the invaluable efforts in this task, from our opinion there is a lack of a global view when using classifiers in ADAS. For instance, there are assumptions that can be made in generic people detection but can/should not be made for PPSs, for example using a scale pyramid or assuming high-resolution targets. In this chapter we study some interesting points related to pedestrian classification that are independent from a specific feature or algorithm. First we introduce a set of requirements for features and learning algorithms in order to be used in any kind of PPS. Then, we present two classifiers, one based on the so-called histograms of oriented gradients and another based on Haar features and edge orientation histograms. Then, a set of experiments is performed aimed at answering several questions like the effect of foreground segmentation on classification and the importance of the source of learning samples to the classifier performance.

The chapter is organized as follows. Section 4.1 overviews the evaluation methodoly used in this chapter. In Sect. 4.2 we describe the requirements for features and learning algorithms. Section 4.3 presents two classifiers that fulfill the proposed requirements. Section 4.4 presents the experimental results, which are analyzed throughout the discussion carried out in Sect. 4.5.

## 4.1 Evaluation methodology

In this chapter we use the CVC-02-Classification dataset (App. B). The training procedure has been carried out using the training subset, first with the provided cropped positives and negatives, and then one bootstrapping iteration has been performed by collecting 10 000 hard negatives, that is, false positives provided by the first trained classifier. For the testing we use either the window-based and image-based testing subsets depending on the curve to be plotted. Although there exist other recent ADAS datasets, we bet for the performance testing with our proposed dataset for three reasons. First, in order to make use of the same data source (CVC-02) along the thesis. Second, in order to perform classifiers comparisons with different foreground segmentation algorithms, we need frames with 3D data, and just CVC-02-Classification provides it. Third, the volume of data is more adjusted to the number of experiments to be performed and our computational resources than with larger datasets like DC-02 or Caltech. However, as can be seen in App. B, the variability in CVC-02 is similar to the other datasets so the conclusions extracted from the first are highly extrapolable to the others.

Regarding the evaluation protocols, we use both window-based (FPPW) and image-based (FPPI) plots, taking multiple detections for each annotation as just one true positive since there is not a clustering step involved. Please refer to App. D for more details.

## 4.2 Features and learning algorithms requirements

In Chap. 3 we have introduced new algorithms to generate candidate windows that reduce the number of potential false positives (non pedestrian candidates) while also reduce the number of windows to be classified. One of the promising algorithms is Adaptive Road Scanning, which focuses the generation on the important target of the system, i.e., the road. As has been demonstrated, the reduction in the number of candidates is around a 98% with respect to the sliding window approach. However, since these approaches drive the classifiers input, the use of some ad hoc classification optimizations is restricted. For instance, the scale pyramid scheme is not appplicable, as we will detail later. Moreover, the requirements of the system (e.g., a detection range from 0 to 50m) also affect the characterisics of the classifier.

Attending to the system demands and foreground segmentation approach, the requirements of the features and classification algorithms can be summarized:

- Classifiers must be able to deal with candidates of unrestricted scales. The first idea that comes into one's mind is to rescale each window to the size required by the classifier. This is obviously inapplicable even in toy experiments. The second idea is to construct a scale pyramid compatible with the Adaptive Road Scanning in order to use fixed window sizes, which is the requirement of for example, histograms of oriented gradients [35]. However, the task of trying to match candidates of unrestricted size in the pyramid not only seems to add unnecessary complexity but also has two practical problems. The first is a scale resolution loss, illustrated in Fig. 4.1. In order to have a similar number to the window scales provided in the adaptive road scanning the pyramid should have twice the number of scales than the original pyramid with a scale step of $\Delta = 1.05$ can provide. As can be also appreciated, if a single scale for scanning position was used then the pyramid would be feasible, but of course this would results in a drop of the candidate generation performance. The second problem is not focused on the number of scales but on their computation cost in terms of time and space. If a 64 × 128 window is used for the classification as in [35], then the different pyramid scales must be rescaled to cope with the different real scales of the pedestrians in the original image. In the case of ADAS the aim is to classify windows as small as 12 × 24, that is, at a distance of 50m, which means having to upscale the original image by 5, ending up with a scale of 3400 × 2560 pixels for this case. This matter has been actually tested and the system becomes significantly slowed down. Hence, the approach must be focused on developing classifiers not constrained to an specific candidate window size but able to deal with arbitrary candidate positions and sizes in the image.

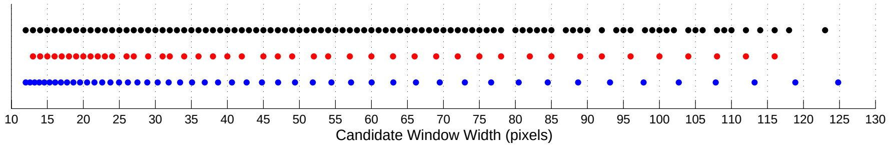


**Figure 4.1:** Width of the candidate windows provided by (*bottom-row, blue*) pyramid-based sliding window [35], (*middle-row, red*) Adaptive Road Scanning with a single pedestrian size ($0.85 \times 1.70m$) and (*top-row, black*) Adaptive Road Scanning with variable pedestrian size ($\mathcal{S} = 5$, as introduced in Chap. 3). Each width corresponds to one different scale, and in the case of a scale pyramid to a scaled image. For example, if we take the windows with widths between 80 and 90 pixels, it can be seen that the adaptive with variable pedestrian size contains around ten different sampling scales while the other two techniques just test three scales.

- Candidates can be placed in any position of the image, as a result of applying a focused denser scan on certain areas (e.g., adaptive road scanning). This denser scan is better than a fixed-step scanning in the sense that it adjusts the candidates to the pedestrians at finer steps than a the fixed-step scanning ones, which can potentially improve the classification results. The drawback in this case is that the precomputation optimizations made by cell-based features (e.g., HOG) is not possible. However, as will be seen in this chapter this kind of precomputation is just possible in a few specific cases.

- They must adapt to a large working range, i.e., windows from 24 to 200 pixels high. Figure 4.2 simulates the variability of the pedestrian at different distances.

- Since the number of windows to be processed is high, the cost assumed to computing features shall be low, which means that the use of pixel-wise operations when computing each feature should be avoided.

- Although it can usually be assumed that all the candidates keep the same aspect ratio, this is not necessarily true and in fact our experience says that we can expect just the contrary for future research (e.g., pedestrians have many varied aspects ratios depending on age or body complexion).

- In our proposed system, the addition of a margin to the candidates is responsibility of the classifier, so it must be able to deal with its inconveniences, for example candidates going out of the image.

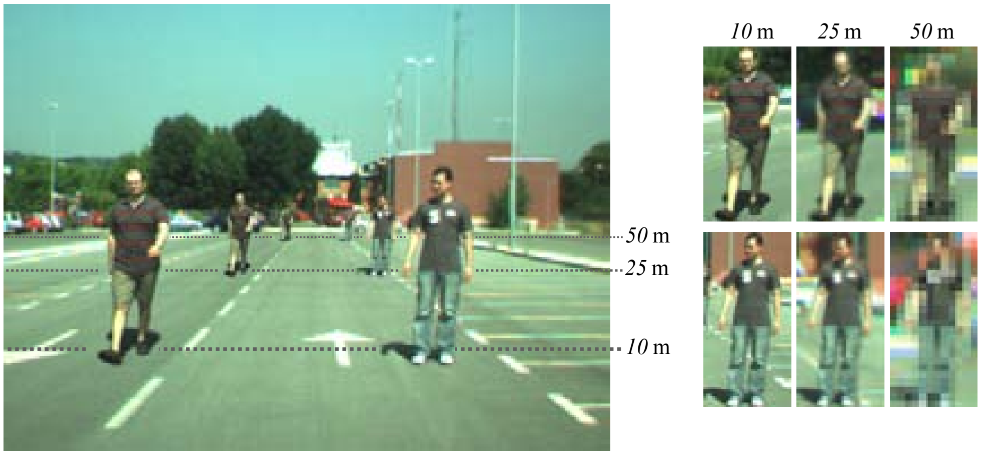


**Figure 4.2:** Variability of pedestrian appearance according to distance (simulated).

## 4.3 A study of classifiers

Once the properties of features and classifier have been defined, we propose two types of classifiers that adjust to them and then we analyze interesting aspects regarding foreground segmentation, training, etc.

### 4.3.1 Simplified histograms of oriented gradients with SVM

Histograms of oriented gradients (HOG) [35], proposed by N. Dalal and B. Triggs in 2005, have become a performance baseline in human detection literature. HOG are SIFT-inspired features [99] that rely on local gradient orientation information. The idea is to divide the incoming window into small cells (boxes in the Fig. 4.3) forming a grid. Cells are grouped in bigger spatial regions $R$ called blocks (white box in the same figure). A histogram is computed for each block consisting in $\beta \times n$ bins, where $\beta$ is the number of gradient orientations and n is the number of cells in the block. Concretely, each pixel magnitude is added to the corresponding pixels orientation by using trilinear interpolation, that is, votes are interpolated both along neighboring cells and neighboring orientation bins. Then, the histogram is normalized (the authors suggest L2-norm) to define the final feature vector attached to each block. The feature vectors are fed to a *support vector machine* to construct the model.

According to the recent studies by Dollár et al. [41] and Enzweiler et al. [43], HOG with SVM stands as one of the best pedestrian detection algorithms for ADAS. However, these features do not fulfill the previous list of requirements: they require a scale pyramid and have a high computational cost since they perform pixel-wise operations for each feature.

We propose a simplified version of HOG that we call *simplified histograms of* *oriented gradients* (SHOG), which are able to work with any window size (i.e., the scale pyramid is not needed) by adding two modifications to the original descriptor:

- avoid the use of a Gaussian mask to weight the pixel magnitudes with respect to the cell center.
- introduce the use of the integral image to store the magnitude in each orientation bin, which restricts us from using the bilinear interpolation in space (orientation interpolation is kept).

The SHOG descriptor (Fig. 4.3), which can have the same number of blocks, cells and orientations than HOG, is fed to the SVM classifier as in the original proposal. The descriptor is introduced in [175] to accelerate the computation of original HOG in human detection.

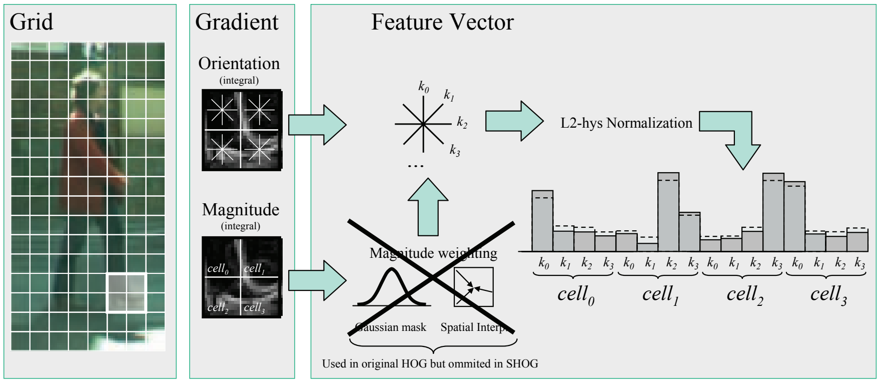


**Figure 4.3:** Scheme of the computation of SHOG features.

Once the version of HOG able to deal with any window size has been described, the first issue to analyze is the difference in performance with respect to the original descriptor. Figure 4.4($a$) displays the performance of HOG and SHOG in INRIA dataset. There exists a performance loss of 6% of SHOG with respect to HOG produced by the descriptor simplification at FPPW= $10^{-4}$, and a loss of 7% of SHOG working with original size windows with respect to SHOG using the pyramid. Hence, in INRIA dataset, the price of using a classifier able to work with multiscale candidates is a 13% loss in classification performance. Figure 4.4($b$) shows the same experiment but using an ADAS specific dataset, i.e., CVC-02-Classification. In this case, SHOG with not resized samples outperforms both the pyramid-based approaches: a gain of 5% with respect to HOG and a 9% with respect to SHOG with the scale pyramid at FPPW=$10^{-4}$.

At first sight it can be surprising that the classifier that fulfills the aforementioned requirements also provides a boost in performance with respect to the pyramid based one, even with respect to the not-simplified original HOG. However, in the end the explanation is quite intuitive. In the case of INRIA dataset, the original pedestrians are bigger than 128 pixels high, so when downscaling them to the 64 × 128 pixels canonical scale the details of such a high-resolution pedestrians are not lost. Hence, in such an scenario, there are two outcomes. First, rescaling the candidates provides a favorable *normalization* of the image details, which makes the corresponding classifiers work better. Second, the original HOG takes great advantage of the details, so when applying the simplifications there is a performance loss.

On the contrary, ADAS dataset contains both high- and low-resolution pedestrians. As an example, the canonical INRIA sample of $64 \times 128$ pixels would correspond to a pedestrian at about 10 m in our system, while CVC-02 contains pedestrians in the range from 0 to 50 m, i.e., from $70 \times 140$ to $12 \times 24$ pixels. In this context, where scale variability is high and details are often not well-defined, it is better to let the learning algorithm work with *unnormalized-to-scale* features.

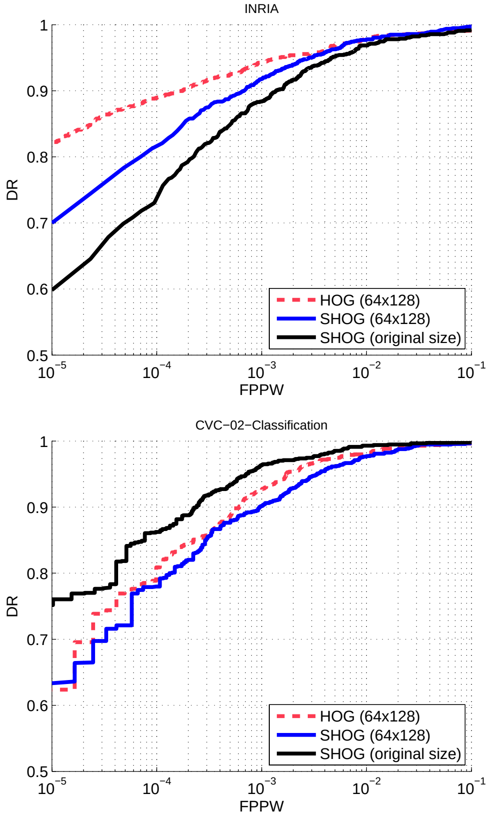


**Figure 4.4:** HOG (using pyramid) vs SHOG (using pyramid and original image) performance in INRIA and CVC-02 datasets.

### 4.3.2 Haar features and edge orientation histograms with Real AdaBoost

Haar features (HF) are simple and fast-to-compute features, reminiscent of Haar basis functions used by Papageorgiou et al. [124] for object detection. A feature of this set is defined by a filter that computes the gray level difference between two defined areas (white and black):

$$\text{Feature}_{Haar}(x, y, w, h, type, R) = E_{white}(R) - E_{black}(R) \ ,$$

where $x, y$ is the bottom-left position of the given image region $R$; $w, h$ represent rectangle width and height; *type* is one of the filter configurations listed in Fig. 4.5*(bottom)*, and $E_{area}(R)$ is the summation of the pixels in the region area (white or black). In order to compute $E$, the *integral image* (*ii*) representation [159] has been used, in which the summed values of a certain region can be efficiently computed by four *ii* accesses. See Fig. 4.5 for an schematic illustration.

Edge orientation histograms (EOH) are proposed by Levi and Weiss for face detection in [95]. They rely on the richness of edge information, so they differ from the intensity area differences of Haar features but maintain the same invariance properties to global illumination changes. First, the gradient is computed by a Sobel mask convolution (contrary to the original paper, no edge-thresholding is applied in our case). Then, gradient pixels are classified into $\beta$ images (in our case we have tested $\beta = \{4, 6, 9\}$) corresponding to $\beta$ orientation ranges (also referred as *bins*). Therefore, a pixel in bin $k_n \in \beta$ contains its gradient magnitude if its orientation is inside $\beta_n$'s range, otherwise is null. Integral images are now used to store the accumulation image of each of the edge bins. At this stage a bin interpolation step has been included in order to distribute the gradient value into adjacent bins. This step is used in SIFT [99] and HOG [35] features, and in our case it slightly improves the EOH performance. Finally, the feature value is defined as the relation between two orientations, $k_1$ and $k_2$, of region $R$ as:

$$\text{Feature}_{EOH}(x, y, w, h, k_1, k_2, R) = \tfrac{E_{k_1}(R)+\epsilon}{E_{k_2}(R)+\epsilon}.$$

If this value is above a given threshold, it can be said that orientation $k_1$ is dominant to orientation $k_2$ for $R$. The small value $\epsilon$ is added to the factors for smoothing purposes. Please, check Fig. 4.5 for an illustrative overview of these features.

The input window is densely scanned by all possible Haar and EOH features, which are fed to Real AdaBoost learning algorithm given that their number is too high to be managed by other classification algorithms like SVM. In our experiments we omit the use of the so-called cascade of classifiers usually employed for AdaBoost given that it is basically used for optimizing the computation time but does not provide any improvement in detection rates. Hence, we will use just one cascade trained with one bootstrapping iteration, like SHOG+SVM.

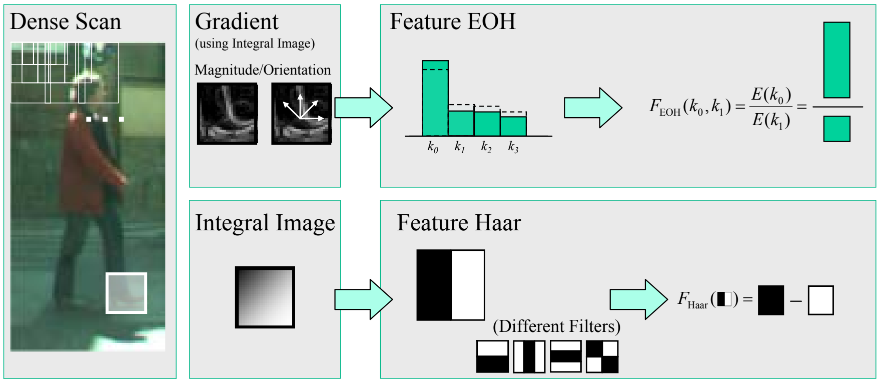


**Figure 4.5:** Scheme of the computation of HaarEOH features.

Fig. 4.6 illustrates the performance of HaarEOH+AdaBoost using CVC-02 dataset. Although HaarEOH+AdaBoost has lower performance than SHOG+SVM (87% vs 77%), it still provides staisfactory performance, just a 3% lower in DR at FPPW= $10^{-4}$ than the original HOG+SVM. In the first training with the provided fixed negative samples set, the performance is really low for the 500-, 2000- and 4000-features classifiers, about a 10% DR at FPPW= $10^{-4}$. With the bootstrapping iteration the performance is increased to the performance shown in the figure. However, as can be seen, increasing the number of weak rules (features used by AdaBoost) more than the 500 does not translate into a performance improvement, just the contrary: there exists some overfitting that slightly reduces the detection rate. Accordingly, for the next experiments the classifier based on 500 features is used.

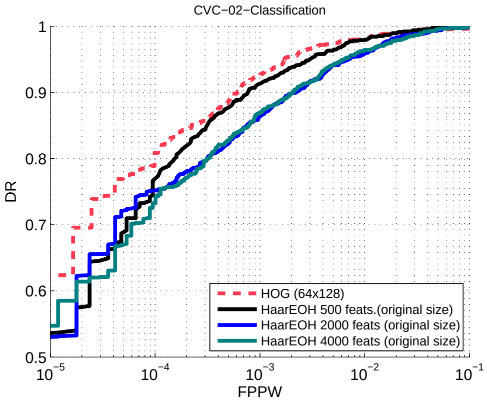


**Figure 4.6:** Scheme of the computation of HaarEOH features.

## 4.4 Experimental results

In the literature, the classifiers performance is typically tested using window-based evaluation, or in the best of the cases, using image-based evaluation by making use of sliding window. In the following study, in addition to these evaluation procedures, the classifier is combined with different foreground segmentation algorithms in order to analyze their combined performance. In addition, we are also interested in their performance in the different risk areas, which has been also shallowly investigated.

Figures 4.7 and 4.8 illustrate the performance of the classifiers when combined with different foreground segmentation algorithms[1]. For the sake of clarity, in the previous plots just one sliding window configuration is plotted, so in order to complete the results Figures 4.9 and 4.10 display the performance of all the configurations.

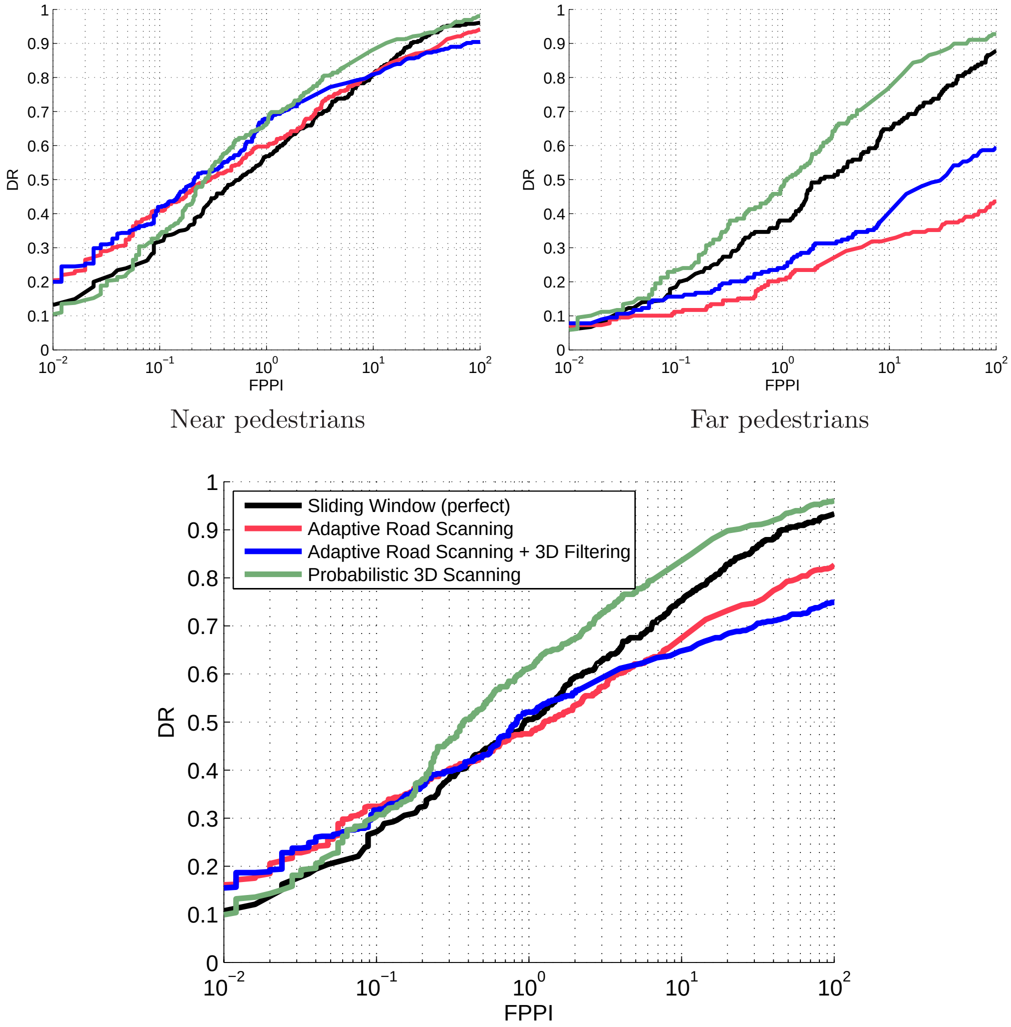
With the aim of going deeper on the analysis, we attach a set of tables (4.1- 4.6) detailing the intersection between the TP, FN and FP sets for each foreground segmentation algorithm. These tables are used to test whether the overlapping between results of different algorithms follows the intuitive trends or not, and are computed by making use of the threshold when imposing FPPI= $10^0$ for each classifier and foreground segmentation algorithm combination. The tables are read by picking one foreground segmentation algorithm on the left column and then checking what proportion of the corresponding set is also present in one of the other algorithms.

Along the experiments we have made use of negative training samples from different sources depending on the classifier: from all the image (sliding windows) and from the road area (road-based algorithms). This is motivated by the idea that by restricting the training negatives examples to the most similar to the testing ones would improve the performance. Figure 4.13 illustrates the performance of a classifier trained using either of the aforementioned negative samples sources and tested using adaptive road scanning. As can be appreciated, contrary to what we expected, there is not a significant difference in the performance.

Finally, we provide some statistics regarding computation time. Classifying a candidate with SHOG+SVM takes 0.32 ms for the features and 0.027 ms for the classifier. According to this, the time spent to classify a frame is more than 5 minutes for sliding window *perfect*, 2 minutes for the *dense* and 1 minute for the *sparse*, always discarding the top 1/3 of the image. In the case of the adaptive road scanning it takes about 10 s and if we add the filtering the time is reduced to 2 s per frame. The time for the probabilistic 3D scanning is also around 10 s. As can be appreciated, the reduction from the fastest sliding window and the slowest of the road-based scans is of one order of magnitude. In the case of HaarEOH+AdaBoost, 500 features are computed in 0.23 ms and the classifier decision in 0.013 ms, a 25% faster in all the cases, so we will ommit the rest of the statistics. The preprocessing needed to compute gradients, integral images, etc. represent a small proportion of the previous numbers.

Figures 4.11 and 4.12 provide qualitative results of the classification.

[1] In order to provide the most fair and realistic comparison, in the case of the sliding window versions we omit the upper 1/3 of the image as it is suggested in Chap. 3.

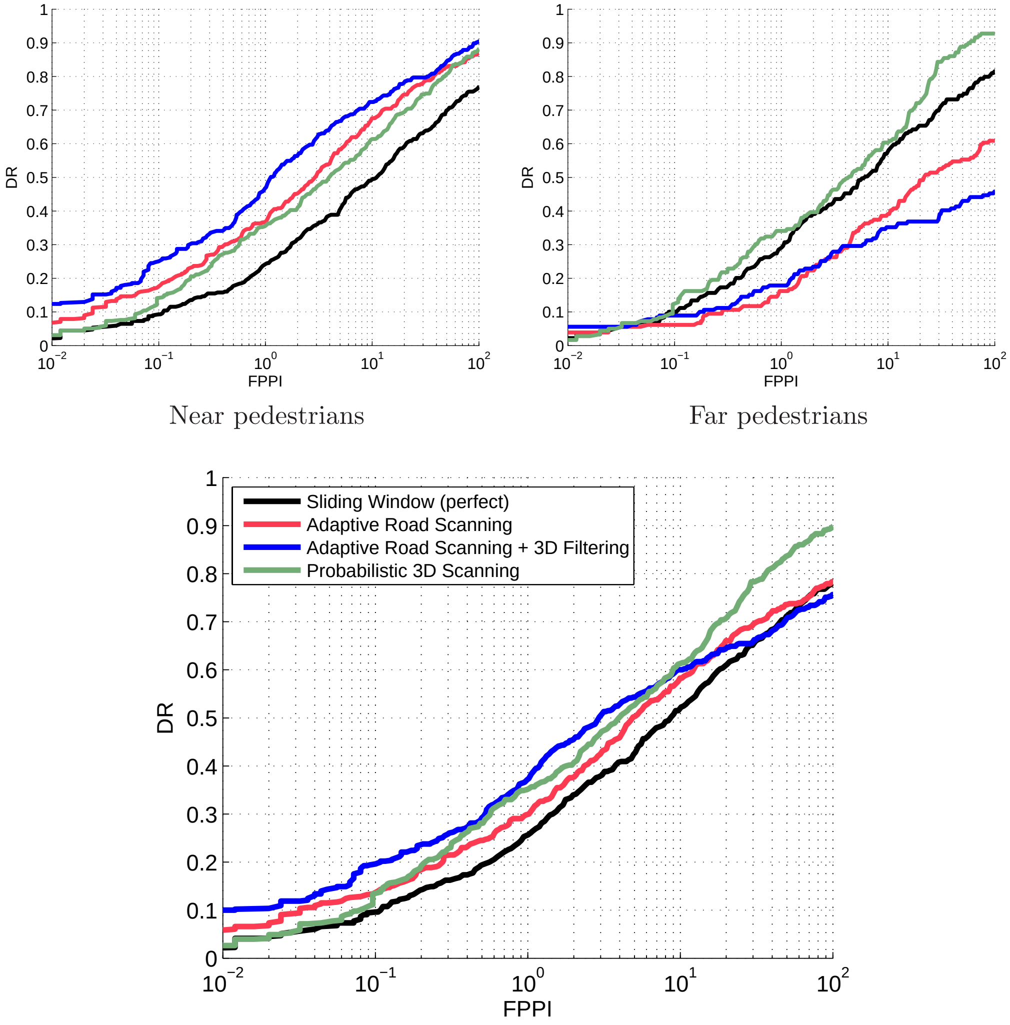


Near pedestrians

Far pedestrians

Whole range

**Figure 4.7:** SHOG+SVM performance with different foreground segmentation algorithms.

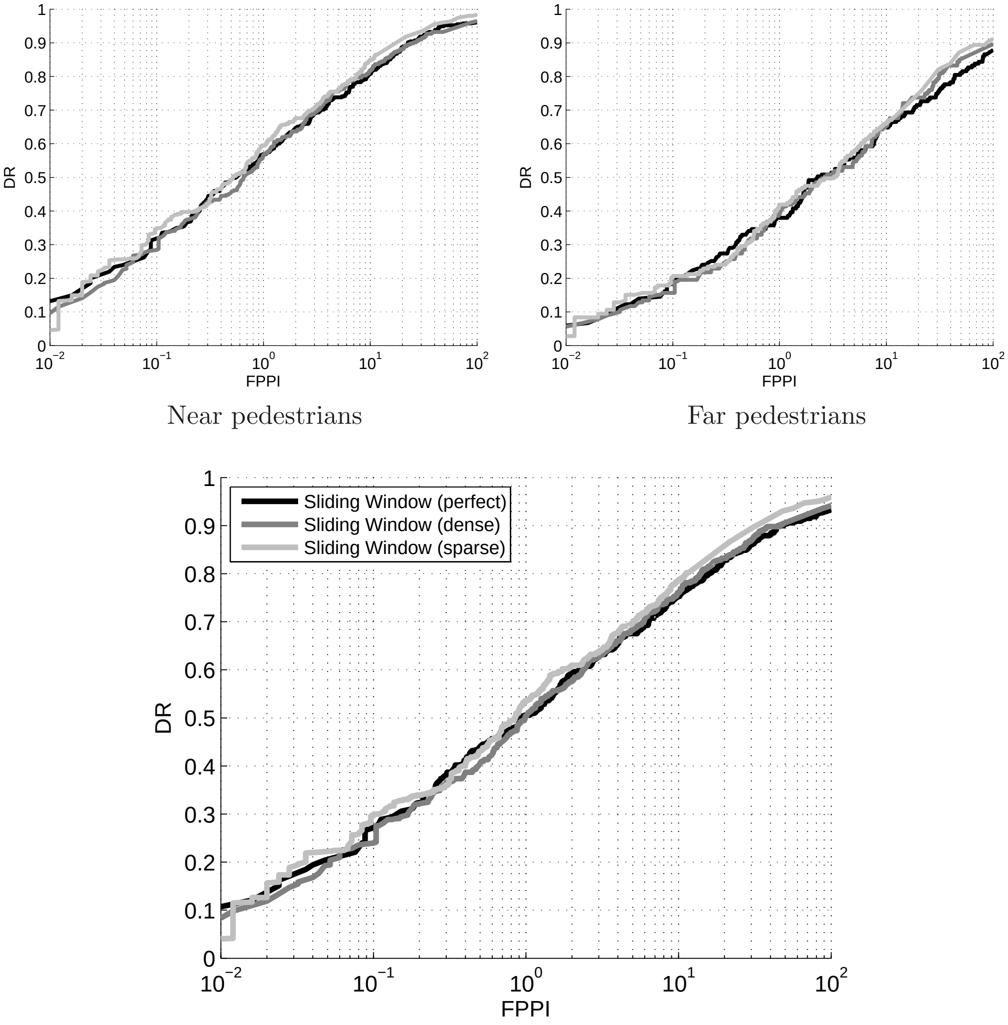


Near pedestrians

Far pedestrians

Whole range

**Figure 4.8:** HaarEOH+AdaBoost performance with different foreground segmentation algorithms.

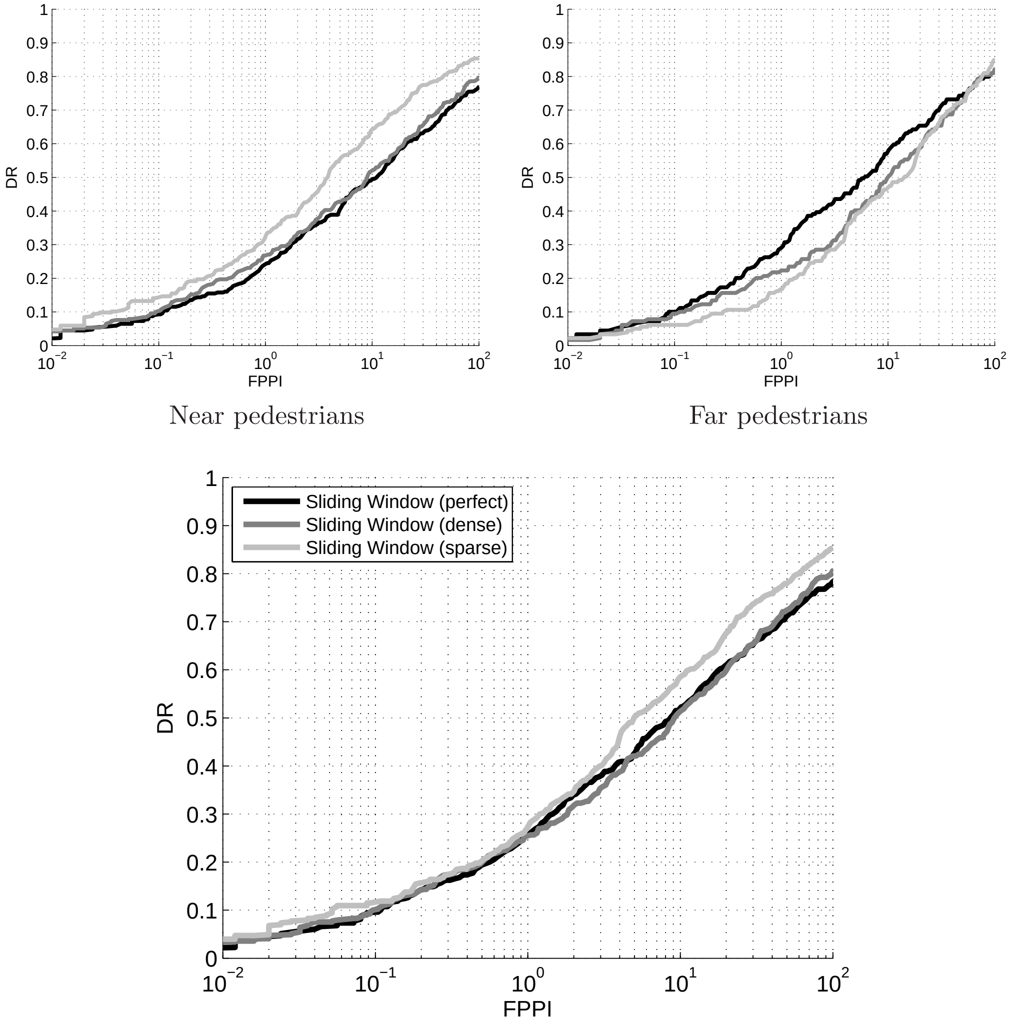


Near pedestrians

Far pedestrians

Whole range

**Figure 4.9:** SHOG+SVM performance with three sliding window configurations.



Near pedestrians

Far pedestrians

Whole range

**Figure 4.10:** HaarEOH+AdaBoost performance with three sliding window configurations.

**Table 4.1:** True positives sets intersection (SHOG+SVM).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 91% | 89% | 75% | 71% | 89% |
| Sliding (dense) | 92% | 100% | 87% | 72% | 69% | 88% |
| Sliding (sparse) | 83% | 82% | 100% | 71% | 69% | 89% |
| Adaptive | 79% | 76% | 80% | 100% | 87% | 85% |
| Adaptive+3D | 70% | 68% | 71% | 81% | 100% | 80% |
| Probabilistic3D | 73% | 72% | 77% | 66% | 67% | 100% |

**Table 4.2:** True positives sets intersection (HaarEOH+AdaBoost).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 68% | 72% | 64% | 61% | 80% |
| liding (dense) | 67% | 100% | 65% | 61% | 56% | 78% |
| Sliding (sparse) | 66% | 61% | 100% | 65% | 66% | 76% |
| Adaptive | 53% | 52% | 59% | 100% | 87% | 76% |
| Adaptive+3D | 42% | 39% | 49% | 71% | 100% | 61% |
| Probabilistic3D | 55% | 55% | 57% | 62% | 61% | 100% |

**Table 4.3:** False negatives sets intersection (SHOG+SVM).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 92% | 81% | 80% | 68% | 67% |
| Sliding (dense) | 91% | 100% | 80% | 77% | 66% | 66% |
| Sliding (sparse) | 87% | 86% | 100% | 79% | 68% | 71% |
| Adaptive | 86% | 73% | 70% | 100% | 80% | 61% |
| Adaptive+3D | 70% | 68% | 66% | 88% | 100% | 58% |
| Probabilistic3D | 85% | 84% | 84% | 82% | 82% | 100% |

**Table 4.4:** False negatives sets intersection (HaarEOH+AdaBoost).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 90% | 89% | 92% | 72% | 79% |
| Sliding (dense) | 90% | 100% | 87% | 81% | 71% | 79% |
| Sliding (sparse) | 91% | 88% | 100% | 83% | 75% | 79% |
| Adaptive | 87% | 87% | 87% | 100% | 85% | 82% |
| Adaptive+3D | 85% | 84% | 87% | 95% | 100% | 76% |
| Probabilistic3D | 92% | 92% | 91% | 89% | 78% | 100% |

**Table 4.5:** False positives sets intersection (SHOG+SVM).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 87% | 77% | 11% | 10% | 29% |
| Sliding (dense) | 71% | 100% | 70% | 9% | 7% | 23% |
| Sliding (sparse) | 43% | 54% | 100% | 18% | 14% | 26% |
| Adaptive | 11% | 16% | 42% | 100% | 49% | 37% |
| Adaptive+3D | 10% | 14% | 46% | 72% | 100% | 37% |
| Probabilistic3D | 33% | 36% | 47% | 21% | 22% | 100% |

**Table 4.6:** False positives sets intersection (HaarEOH+AdaBoost).

| | Sliding (perfect) | Sliding (dense) | Sliding (sparse) | Adaptive | Adaptive+3D | Probabilistic3D |
|---|---|---|---|---|---|---|
| Sliding (perfect) | 100% | 55% | 41% | 21% | 9% | 27% |
| Sliding (dense) | 53% | 100% | 37% | 13% | 6% | 27% |
| Sliding (sparse) | 28% | 31% | 100% | 10% | 11% | 17% |
| Adaptive | 33% | 34% | 25% | 100% | 52% | 44% |
| Adaptive+3D | 20% | 14% | 29% | 61% | 100% | 25% |
| Probabilistic3D | 28% | 36% | 28% | 21% | 14% | 100% |

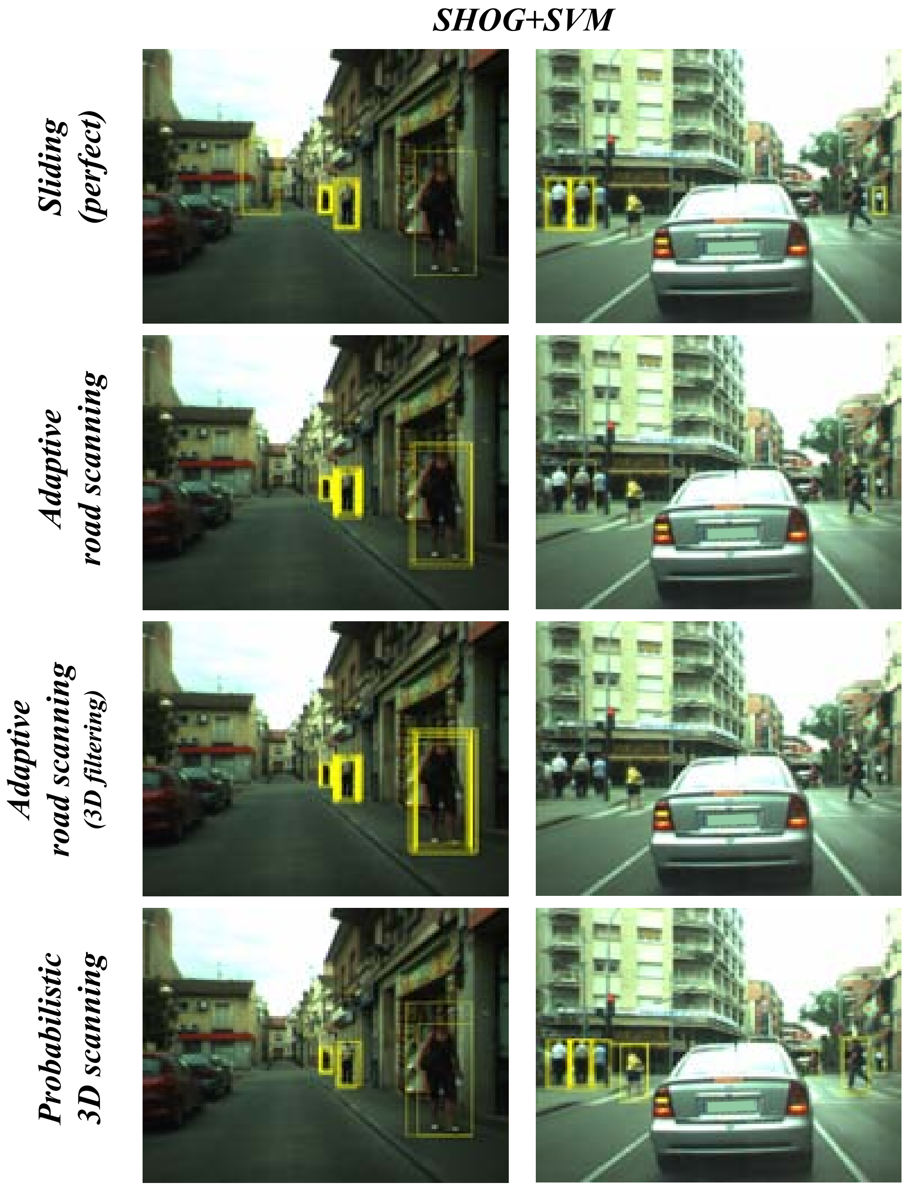

SHOG+SVM
Sliding (perfect)
Adaptive road scanning
Adaptive road scanning (3D filtering)
Probabilistic 3D scanning

**Figure 4.11:** Results after SHOG+SVM classification.

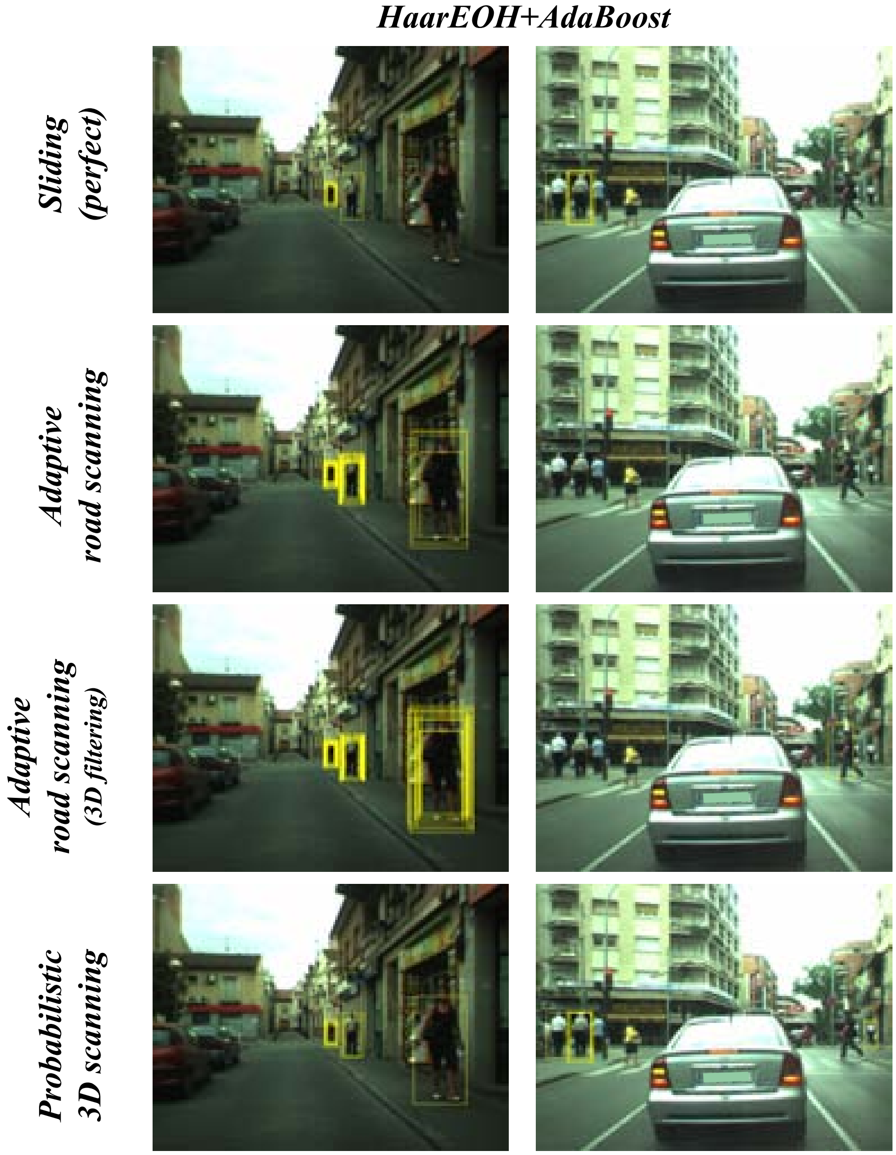
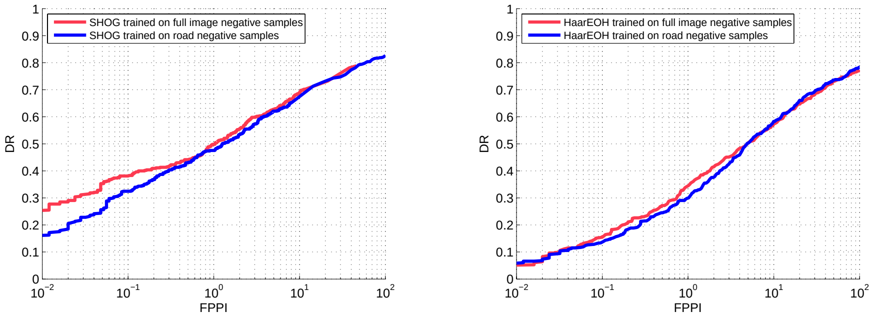

HaarEOH+AdaBoost
Sliding (perfect)
Adaptive road scanning
Adaptive road scanning (3D filtering)
Probabilistic 3D scanning

**Figure 4.12:** Results after HaarEOH+AdaBoost classification.



(*a*) (*b*)

**Figure 4.13:** Detection results depending on the source of negative training samples.

## 4.5 Discussion

By analyzing the experimental results the most important conclusion is that neither the performance of the classifier nor the foreground segmentation can be neglected, each ingredient has its effect on the final system performance. In this section we provide discussion based on the plots and tables previously introduced. We divide the discussion in the following points, each one focusing on a given idea based on a figure or table.

- First we focus on the plots according to the distance range (Figures 4.7 and 4.8). In the case of near pedestrians, sliding window (with any of the parameter configuration) provides worse results than the other algorithms. On the other hand, adaptive road scanning with 3D-based filtering has the best performance in the two classifiers. Both phenomena are related and can be explained by the fact that at short distances the number of potential false positives is low. Even for scanning algorithms that can lose a number of pedestrians as a result of the road estimation error, like the adaptive, the number of wrongly not selected candidates is low, which combined with the also low number of non pedestrian candidates leads to a favorable tradeoff for the road-based schemes with respect to sliding window approaches. In the case of the far distances, the opposite phenomenon is found: purely road-based algorithms (adaptive with and without 3D filtering) lose performance while sliding window maintains the performance. Probabilistic 3D shows its full potential here by not losing any performance at all thanks to the adapted scanning to the uncertainty of stereo and estimated road plane along distance. In fact, this is what makes probabilistic 3D the strongest algorithm: it is able to provide a balanced tradeoff between non pedestrian candidates and detection rate for near pedestrians but also keep the detection rate at far distances even improving in a 5-10% the sliding window in both algorithms at FPPI= $10^0$. The whole range plots represent the combination of the near and far ranges. Although it is difficult to extract a global conclusion here given that each classifier has its different response, some evidences can be pointed out. It can be said that for the case of SHOG + SVM the probabilistic 3D has a clear advantage with respect to the other algorithms, which perform quite similar, whereas in the case of HaarEOH + AdaBoost the difference is not that clear and it is affected by the outstanding performance of adaptive and filtered road scanning in the near distances.

- As can be appreciated, HaarEOH + AdaBoost performs worse than SHOG + SVM in general, which is predictable attending to the window-based plots. However, the effect of this performance decrease is different depending on the foreground segmentation. The clearest example of this can be seen in near pedestrian plots: the performance drop between SHOG and HaarEOH curves is of about a 20% in the case of adaptive with 3D filtering whereas it is of a 35% in the case of probabilistic 3D scanning. The reason is that although HaarEOH loses more pedestrians than SHOG, foreground segmentation algorithms that generate a larger number of candidates affect more SHOG than HaarEOH. We illustrate this effect in Fig. 4.14.

- With respect to the performance of the different sliding window configurations, in Figures 4.9 and 4.10, the differences are not big, in fact in the case of SHOG + SVM it is not significant at all. This case seems quite similar to the one described previously, in fact if we look at the classifier thresholds needed to achieve FPPI= $10^0$, they tend to be higher for the *perfect* version parameters than for the *sparse* version, which should lead to a higher number of false positives, for instance in the sparse version. However, contrary to what we expected, by having a qualitative evaluation of the results we have noticed that the misdetections/false positives are quite balanced when imposing the aforementioned thresholds, which makes us conclude that for the current dataset the *sparse* version parameters are sufficient to achieve the same overall performance as the perfect one, that is, the classifier also plays its role in the system.

- There are some interesting things that can be extracted from the TP, FN and FP sets intersection tables. For instance, the intersection between *dense/sparse* and *perfect* versions is not 100%, which means that what we call *perfect* when selecting the candidates in fact is not that perfect after applying the classifier, specially if we have in mind that the detections are thresholded according to the FPPI= $10^0$ criterion. It is also worth to mention the rate of intersection between all the algorithms with respect to probabilistic 3D, always between 80-90% in the case of SHOG and 60-80% in the case of HaarEOH, which means that the latter algorithm is capable of detecting a high number of the pedestrians detected by the other algorithms. It can also be seen that HaarEOH is less stable than SHOG, that is, not only FPPI plots are lower but also the intersection between hits is also lower, which up to some extent is reasonable given that it seems to be the less precise classifier of the two. If we look at the false negatives, Tables 4.3 and 4.4, we can see that not only the sliding window versions tend to have a high overlapping but also the road-based candidate generation algorithms, which means that the misclassified pedestrians by, for example probabilistic

3D, are also lost by sliding window and vice versa. The last two tables are the ones that provide more insights on what is happening with the classification. Tables 4.5 and 4.6 detail overlapping in false positives sets, in which, as can be seen, the numbers are much lower. As expected, the highest intersections are present in the sliding windows given that they cover all the possible regions, even though the density is different (which makes the numbers not be 100 but, e.g., 50). In the case of adaptive road scanning, half of the false positives are also present when incorporating the 3D filtering. On the complementary case, a higher number of the false positives when using the 3D filtering version are also present in the raw adaptive version, which follows the logic given that one set is a subset of the other. One shall expect a 100% of intersection in this case, however we shall remind that the threshold used in the classifiers is different as it is selected to achieve a FPPI= $10^0$.

- Regarding the source of the negative samples, it has been mentioned that the difference is not significant. This is presumably as a result of the bootstrapping iteration, so we could have indistinctly used either of the two options that the conclusions of the previous study would have been the same.

- One final remark that is worth to mention with respect to computation time is the topic of feature precomputation. It represents an interesting way of boosting the computation speed that is used in some papers. In the case of [35], given that the sliding window step matches the features cell step, it is possible to precompute all the feature cells of the image and then access to their values when the candidate classification requires it. The precomputation is potentially useful for the techniques in [133, 102, 51] and for the *perfect* version of sliding window algorithm of this thesis. For example, in this case a time reduction of 40% is achieved according to our experiments with SHOG and the *perfect* sliding window parameters. However, the use of percomputation is very restricted to specific features and candidate generation algorithms. As an example, several techniques in which features do not follow a regular grid will not be able to exploit this [124, 112, 141, 66, 114, 68, 155, 167, 138, 175, 123, 152, 40, 96]. Furthermore, even when there exists a regular grid of features like with HOG, there is a strong requirement in the sense that the features grid shall fit the candidates scanning step, which does not apply, for example in [44] and in the *sparse* version of sliding window.

Finally, we point out some points that represent promising lines of research with respect to classification, both directly related and also unrelated to the study carried out in this chapter:

- Having seen the difference in performance rates depending on the distance range, the next intuitive research step is to specialize the system on these two different ranges. On one hand to train a classifier with samples from near pedestrians and another with far pedestrians, which should add robustness to the classifier, similar to the idea of using different classifiers depending on the distance suggested

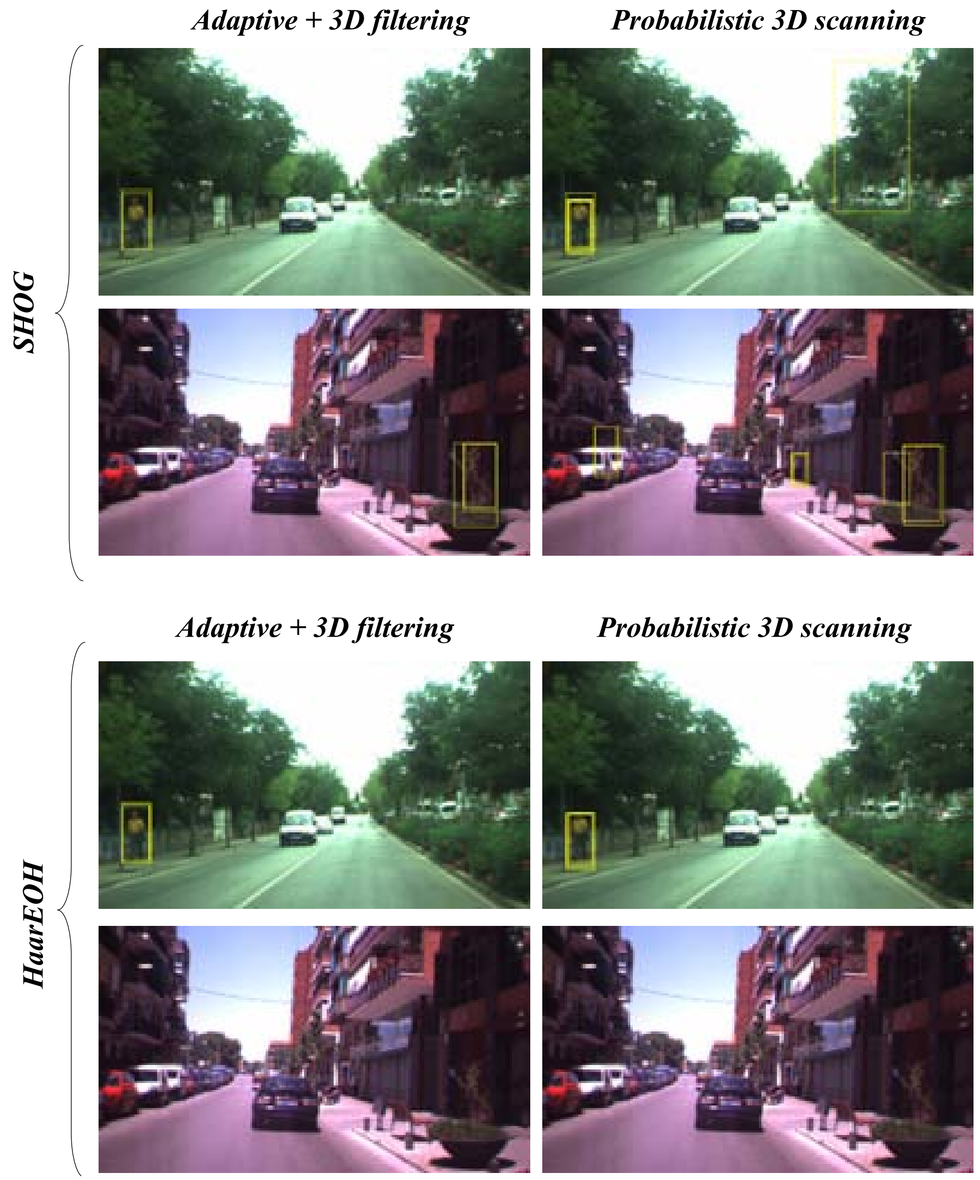


**Figure 4.14:** Different effect on two different foreground segmentation algorithms depending on the classifier performance.

by Enzweiler et al. in [44]. On the other hand use different foreground segmentation algorithms according to the distance, for instance adaptive road scanning with 3D filtering for the near range, attending to its satisfactory performance, and adaptive road scanning or probabilistic 3D for the far pedestrians.

- Regarding generic pedestrian classification, new features, learning algorithms and classification paradigms are always likely to outperform the current ones In Chap. 2 we describe some of the newest approaches. However, it is also worth to remark that in order to optimize the system and be able to use the studied foreground segmentation algorithms, the new features and algorithms must fulfill the list of requirements of Sect. 4.2. Finally, we can highlight that multipart and multiclass paradigms are topics of increasing interest for future work in PPS, attending to the literature.

## 4.6 Summary

In this chapter we have studied the requirements of the features and classification algorithms to be used according to the system demands and foreground segmentation approaches. Then we have presented two classifiers that fit the previous requirements: simplified histograms of oriented gradients with SVM (SHOG+SVM) and Haar features and edge orientation histograms with Real AdaBoost (HaarEOH+AdaBoost). Then, we have studied the performance of these classifiers independently using an image-based evaluation and combined with different foreground segmentation algorithms using an image-based evaluation.

*Down and Up*
Barbara Kirsch, oil on canvas, 2009
(Private Collection)

The relation of *Up and Down* and the chapter is straightforward: silhouettes. Contrary to the typical urban scenes, in which clutter, clothes and illumination make silhouette extraction hard even for human beings, the painting depicts two ideal human shapes that are very rarely found in real life. However, the scene indeed contains some details that in fact make shape extraction difficult even in this case. These details, like carried bags or the shadows, are typically unnoticed but suddenly pose a problem when developing algorithms since it is hard to provide a frontier between the pedestrian and the non-pedestrian regions.

# Chapter 5

# Detections refinement

As described in Chap. 2, verification and refinement have received limited attention. The verification methods in the literature are very tied to the classifier response, that is, the verification is mostly focused on the type of false positives the classifier gives (e.g., trees, vertical objects, etc.). Hence, it is difficult to make a global study on the most reliable verification techniques. In the case of the refinement, although the algorithms also depend for example on the classifier, it seems more plausible to make a performance comparison since they are likely to be used after most of the existing classifiers. Accordingly, in this chapter we are going to focus just on refinement, concretely on two relevant aspects that have received few attention. First, in Sect. 5.1 we propose a new clustering technique to group detections, which together with the well-known mean shift algorithm, one of the few proposed in the literature, will serve as starting point for a global study that takes into account classifier precision and foreground segmentation techniques. Second, attending to the lack of literature on unsupervised shape extraction (Chap. 2), we aim to cover the gap by proposing a novel silhouette refinement technique that does not require explicit shape annotation for training (Section 5.2). In this chapter, given that the two topics studied are independent, the discussion is included in each section instead of making a single one as in previous chapters.

## 5.1 Detections clustering

The proposals on detection clustering for human detection have been limited for years to the algorithm proposed by N. Dalal [34]. In this section we also propose a fast clustering algorithm that consists in detections accumulation based on overlapping and two possible distance criteria. After formalizing the algorithms, we aim to perform a set of experiments with the aim of answering some relevant questions: *Is there a significant difference between the presented algorithms, or on the contrary they can be interchanged without a significant performance drop? How is the clustering affected by the precision of classifiers? How much weight does foreground segmentation have in the clustering? Is there such an overall best clustering algorithm and parameters configuration? Is there a difference in computation time?*

### 5.1.1 Mean shift clustering

Mean shift, proposed by D. Comaniciu [33], is a clustering algorithm that iteratively finds the modes of a density function. The use of mean shift for windows clustering after pedestrian classification is proposed by N. Dalal [34]. The main steps of Dalal's proposal are detailed in Algorithm 7.

Note that this algorithm, together with our proposed one, makes use of the windows scale $s$. Given that we do not make use of the scale pyramid, the scale of the detections is set to $s = c_h/d_h^m$, where $c_h$ is the height of the window and $d_h^m$ is the height of the minimum detection window, for example 24 pixels in our case.

**Algorithm 7** Mean shift Clustering

```
Input:
  A set of detection windows, each with coordinates $c = (c_x, c_y, c_s)$.
  $\sigma_x$, $\sigma_y$ and $\sigma_s$ are the smoothing values that control the uncertainty of a
    detection.
Initialization:
  Create a set of points $\mathbf{p_i} \in P$, where each $\mathbf{p_i} = [x_i, y_i, s_i] = [w_x^i, w_y^i, \log(w_s^i)]$
    and $c_i$ is the confidence of $w^i$.
  Attach an uncertainty matrix $\text{diag}(\mathbf{H_i}) = [(\exp(s_i)\sigma_x)^2, (\exp(s_i)\sigma_y)^2, (\sigma_s)^2]$
    to each point.
Algorithm:
  While $P$ has points
    Pick a random $\mathbf{p_i}$ and assign it to current mean $\mathbf{y}$
      Do until $\mathbf{y}$ converges.
        For each $\mathbf{p_i} \in P$, compute the point weight
          $\varpi_i(\mathbf{y}) = \frac{|\mathbf{H_i}|^{-1/2} t(w_i) \exp(-D^2(\mathbf{y},\mathbf{p_i},\mathbf{H_i}]/2)}{\sum_{i=1}^n |\mathbf{H_i}|^{-1/2} t(w_i) \exp(-D^2(\mathbf{y},\mathbf{p_i},\mathbf{H_i}]/2)}$
          where $D^2 \left[\mathbf{y}, \mathbf{p_i}, \mathbf{H_i}\right] \equiv (\mathbf{y} - \mathbf{p_i})^T \mathbf{H_i}^{-1} (\mathbf{y} - \mathbf{p_i})$ is the Mahalanobis
          distance between $\mathbf{y}$ and $\mathbf{p_i}$, and $t(w_i)$ is a hard clipping transform
          function that sets all negative detection confidences $w_i$ to 0.
        Compute the new mean $\mathbf{y} = \mathbf{H_h}(\mathbf{y}) \left[\sum_{i=1}^n \varpi_i \mathbf{H_i}^{-1} \mathbf{p_i}\right]$,
          where $\mathbf{H_h}^{-1}(\mathbf{y}) = \sum_{i=1}^n \varpi_i(\mathbf{y}) \mathbf{H_i}^{-1}$ is the weighted harmonic mean of
          the matrices $\mathbf{H_i}$ computed at $\mathbf{y}$.
      Delete points belonging to the current density from $\mathbf{P}$ using $\mathbf{H_h}$.
      Attach $\mathbf{y}$ to the list of clusters, with coordinates and the mean of
        the points confidences.
```

### 5.1.2 Accumulative Clustering

We propose an algorithm based on iterative window accumulation and a given distance measure that requires less operations than mean shift. The algorithm takes a detection $i$ and checks if the distance to the current clusters below a threshold. If so, the

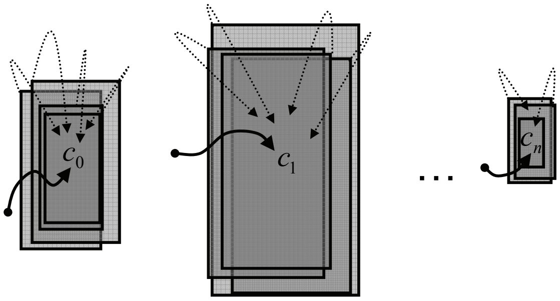


**Figure 5.1:** Mean shift clustering scheme.

detection is added to the corresponding cluster, otherwise, a new one is created. The method is detailed in Algorithm 8.

**Algorithm 8** Accumulative Clustering

```
Input:
  Distance function $D$ and threshold $t$.
Initialization:
  List of detections $\mathcal{D}$ sorted by confidence
  List of clusters $\mathcal{C} \leftarrow \varnothing$
Algorithm:
  For each detection $d_i \in \mathcal{D}$ (step 1)
    attached ← false
    For each $c_j \in \mathcal{C}$
      If $D(d, d_i) < t \; \forall d \in c_j$
        Attach $d$ to $c_j$ and attached ← true
        go to (step 1)
    if attached=false
      Create new cluster $c_{new}$ and attach $d$ to $c_{new}$
    For all $c_j \in \mathcal{C}$
      Compute mean of $d \in c_j$
```

We propose two distance measures: one based on detections overlapping and another based on detections coordinates. The first measure has been widely used for classification algorithms evaluation, as it is explained in App. D. In fact, Enzweiler et al. [44] have recently proposed the use of this measure also for clustering by discarding the lowest confidence detections in a group of overlapped ones. The equation used is the following [47]:

$$D_{overlap}(a, b) = 1 - \frac{\text{area}(a \cap b)}{\text{area}(a \cup b)} \;, \tag{5.1}$$

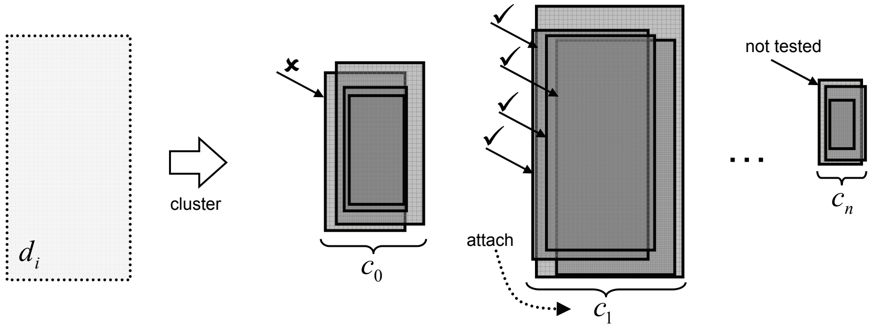


**Figure 5.2:** Accumulative clustering scheme.

where $a$ and $b$ are the detection windows.

The second measure is similar to Dalal's algorithm, since it represents each detection with its coordinates $(x, y, s)$. In our case we do not compute the logarithm of the scale because the scales do not follow an exponential progression. Figure 5.3 illustrates the projection of two clusters of detections into the $(x, y, s)$ space. As can be seen, the $(x, y)$ variance of the clusters is not constant along $S$. The heteroscedasticity of this data poses a problem when trying to compute the distance between points in the sense that the three dimensions cannot be normalized using a simple covariance matrix and Mahalanobis distance. Instead, we propose a distance measure that takes into account the variance with respect to $S$, defined as

$$D_{xys}(a, b) = \frac{\sqrt{(a_x - b_x)^2 + (a_y - b_y)^2 + (a_s - b_s)^2}}{0.5 \cdot (a_s + b_s)} \quad . \tag{5.2}$$

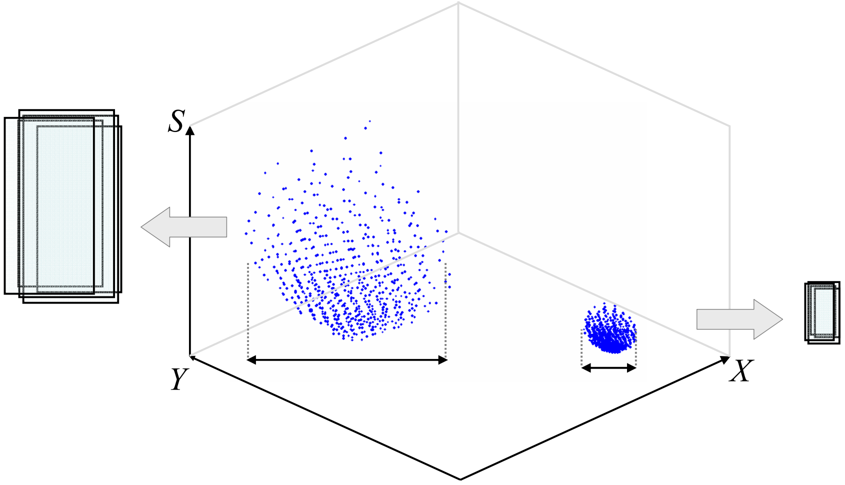


**Figure 5.3:** Projection of the detection windows in the $(x, y, s)$ space.

### 5.1.3 Experimental results

In order to avoid the effect of the classifiers performance on the results we will use two naive synthetic classifiers. Based on the annotations, two classifier simulations are tested according to their localization precision: $\mathcal{S}_{precise}$, in which candidates with an overlapping factor larger than $t = 0.85$ are selected, and $\mathcal{S}_{imprecise}$, in which the threshold is set to $t = 0.5$. Figure 5.4 illustrates both simulations.

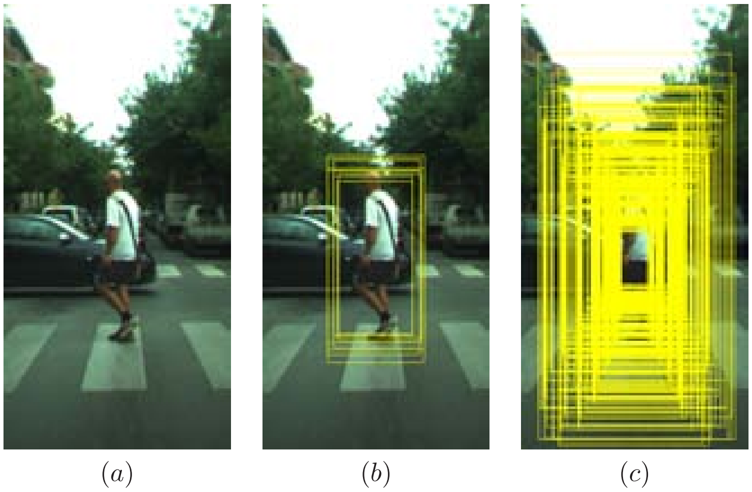


**Figure 5.4:** Simulated Classifiers: $(a)$ Original image, $(b)$ Precise and $(c)$ Imprecise.

Given that after the refinement the selected detection windows are the result of the system, the possible confidence does not take a relevant role, so we will not provide a performance curve as with the classification. Instead, we provide precision and recall based on the resulting raw clusters with no confidence associated and compare results by computing the F-measure[1]. In this case, contrary to the Chap. 4, two detections overlapping one single annotation are counted as one true positive and one false positive.

Tables 5.1–5.8 illustrate the performance of the clustering algorithms on CVC-02-Classification dataset, concretely the 250 frames containing 587 pedestrians used for the image-based evaluation. As foreground segmentation algorithms, we use sliding window (both perfect and sparse configurations are used, dense one is omitted since it does not provide any additional information), adaptive road scanning and probabilistic 3D scanning (these latter with their best parameter configuration described in Chap. 3). The best performance is shaded.

In order to provide an ordered analysis we base it on the questions stated previously:

- *Is there a significant difference between the presented algorithms?* We select the best parameters for each method in each foreground segmentation algorithm

[1] F-measure $= \frac{2\cdot\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

and synthetic classifier, and compute their corresponding f-measure. In all the cases, the difference is smaller than 0.05, i.e., a 5%, but in average it is around 1%. None of the algorithms performance is clearly above the other, there are cases in which accumulative is superior than mean shift and vice versa, so the criterion to select an algorithm will have to be based on different criteria than performance.

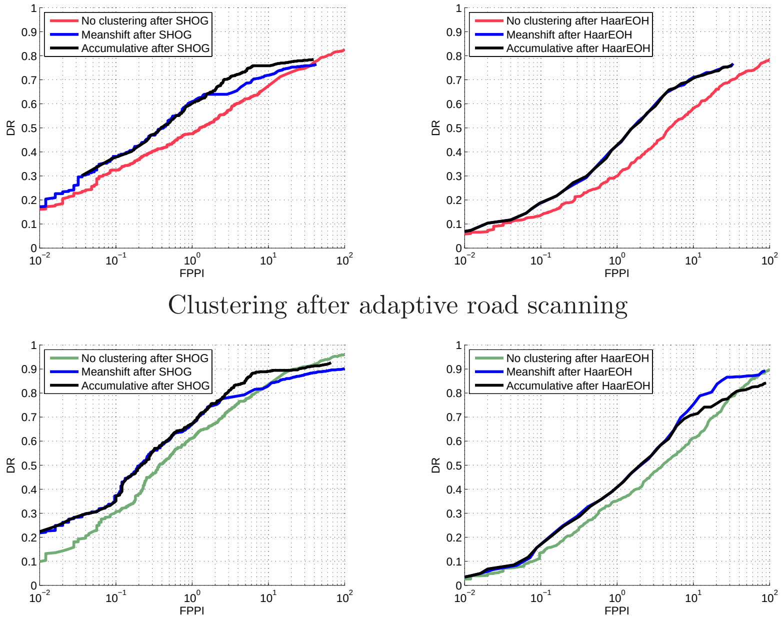
- *How is the clustering affected by the spatial and scale precision of classifiers?* In this case, there is a difference in performance depending on the foreground segmentation algorithm, so this question is very tied to the question *How much weight does foreground segmentation have in the clustering?*. For instance, it can be appreciated that perfect sliding window with a precise classifier provides the best results overall. In this case, if an imprecise classifier is used, the mean shift performance decreases as a result of a higher number of false positives coming from the denser scan (lower precision). The opposite effect can be seen when using the sparse version. In this case, given that the windows density is lower, most of them will not perfectly match with the annotations, so a more imprecise classifier provides a 10% gain in performance. The same behavior can be seen in the adaptive road scanning, that is, a less precise classifier is able to overcome the errors in the candidates generation. With respect to the case of the probabilistic 3D scanning, it appears to be more flexible the two classifiers, presumably because the scan maintains a tradeoff between windows density and their correct placement in the image.

- *Is there such an overall best clustering algorithm and parameters configuration?* As it has been described, there is not such a best algorithm in terms of performance. In the case of accumulative, the overlapping measure, concretely with the threshold set at 0.5, seems pretty robust for all the experiments. With respect to mean shift, although there is not such a clear winning parameter configuration, there seems to be a peak when using $(\sigma_x, \sigma_y) = (4, 8)$, specially when combined with $\sigma_s = \log(3.2)$.

- *Is there a difference in computation time?* In this case there is a clear gain when using accumulative since it is 6 times faster in average than mean shift.

Once the questions have been answered using the synthetic classifiers, an interesting experiment is to check the gain in the system performance when attaching the clustering to two of the relevant foreground segmentation algorithms and classifiers. In order to plot an image-based DR-FPPI curve we take the classifier confidences as in Chap. 4 and then cluster the detections that pass the corresponding threshold by using the previously selected parameters. Figure 5.5 shows improvements of a 15% in DR when using adaptive road scanning and a 5% when clustering after probabilistic 3D scanning at FPPI= $10^0$.

Regarding computational time, both mean shift and accumulative clustering are really fast algorithms compared to the classification stage. In the case of clustering the detections after adaptive road scanning, mean shift requires around 6 ms per frame while accumulative takes around 1 ms. Obviously, these numbers vary depending on



Clustering after adaptive road scanning

Clustering after probabilistic 3D scanning

**Figure 5.5:** Performance comparison between clustering algorithms and classifiers.

the number of detections and their dispersion over the frame, but even with foreground segmentation algorithms with a high number of windows like the sliding window, the computation time does not go beyond the 10 ms. In all the cases, though, the accumulative clustering is faster than mean shift. Figures 5.6 and 5.7 illustrate some results of accumulative clustering.

### 5.1.4 Discussion

As has been seen throughout the experiments, the performance of the two analyzed clustering algorithms is similar both with synthetic and *real* classifiers. As has been demonstrated the selection of the clustering algorithm shall be focused on time comsumption rather than on detection performance, contrary to other components of the system in which the detection performance can greatly vary depending on the algorithm. This is due to the fact that the performance range of clustering algorithms mostly depends on the input detections, so if the parameters are set correctly the performance peaks of the different algorithms should coincide. We have also studied the relation between foreground segmentation and clustering, concluding that clustering detections of precise classifiers not always provides the best performance, it depends on foreground segmentation, for example the *perfect* versus *sparse* versions of sliding window.

**Table 5.1:** Clustering detections after sliding window (perfect) and a precise classifier ($t = 0.85$).

**Mean shift**

| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
|---|---|---|---|---|---|---|
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9248 | 0.8669 | 0.8191 | 0.9229 | 0.8705 | 0.8175 |
| **Recall** | 0.9754 | 0.8113 | 0.7094 | 0.9716 | 0.7867 | 0.6849 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9276 | 0.8052 | 0.7935 | 0.9261 | 0.8556 | 0.7867 |
| **Recall** | 0.9679 | 0.7754 | 0.6452 | 0.9698 | 0.7603 | 0.6264 |

**Accumulative**

| | overlapping | | | xys | | |
|---|---|---|---|---|---|---|
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.9195 | 0.9210 | 0.9187 | 0.4315 | 0.9206 | 0.8849 |
| **Recall** | 0.9924 | 0.9905 | 0.9603 | 0.9924 | 0.9849 | 0.8415 |

**Table 5.2:** Clustering detections after sliding window (perfect) and an imprecise classifier ($t = 0.5$).

**Mean shift**

| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
|---|---|---|---|---|---|---|
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.1907 | 0.2505 | 0.2641 | 0.3279 | 0.3662 | 0.3646 |
| **Recall** | 1.0000 | 0.9886 | 0.8792 | 1.0000 | 0.9792 | 0.8358 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.8163 | 0.8315 | 0.7573 | 0.8317 | 0.8403 | 0.7901 |
| **Recall** | 0.9921 | 0.8943 | 0.7301 | 0.9982 | 0.8245 | 0.6962 |

**Accumulative**

| | overlapping | | | xys | | |
|---|---|---|---|---|---|---|
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.4033 | 0.8856 | 0.8109 | 0.0259 | 0.8581 | 0.7613 |
| **Recall** | 0.9924 | 0.9792 | 0.8905 | 1.0000 | 0.9471 | 0.8245 |

**Table 5.3:** Clustering detections after sliding window (sparse) and a precise classifier ($t = 0.85$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9321 | 0.8891 | 0.8620 | 0.9321 | 0.8898 | 0.8550 |
| **Recall** | 0.7000 | 0.6207 | 0.5660 | 0.7000 | 0.6094 | 0.5452 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9321 | 0.8839 | 0.8480 | 0.9345 | 0.8808 | 0.8475 |
| **Recall** | 0.7000 | 0.6037 | 0.5264 | 0.7000 | 0.6000 | 0.5245 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.9104 | 0.9282 | 0.9321 | 0.7107 | 0.9305 | 0.9283 |
| **Recall** | 0.7094 | 0.7075 | 0.7000 | 0.7094 | 0.7075 | 0.6603 |

**Table 5.4:** Clustering detections after sliding window (sparse) and an imprecise classifier ($t = 0.5$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.2368 | 0.2867 | 0.3000 | 0.4059 | 0.4434 | 0.4325 |
| **Recall** | 1.0000 | 0.9905 | 0.8886 | 0.9981 | 0.9698 | 0.8226 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.8396 | 0.8281 | 0.7770 | 0.8502 | 0.8304 | 0.7900 |
| **Recall** | 0.9981 | 0.8547 | 0.7169 | 0.9962 | 0.8037 | 0.6886 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.0865 | 0.8813 | 0.8359 | 0.0276 | 0.7861 | 0.7709 |
| **Recall** | 1.0000 | 0.9811 | 0.9132 | 1.0000 | 0.9641 | 0.8320 |

**Table 5.5:** Clustering detections after adaptive road scanning and a precise classifier ($t = 0.85$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.8411 | 0.8199 | 0.7935 | 0.9200 | 0.9006 | 0.8443 |
| **Recall** | 0.6094 | 0.5584 | 0.5075 | 0.6075 | 0.5471 | 0.4811 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9469 | 0.9196 | 0.8615 | 0.9497 | 0.9196 | 0.8541 |
| **Recall** | 0.6056 | 0.5396 | 0.4698 | 0.6056 | 0.5396 | 0.4641 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.9393 | 0.9418 | 0.9457 | 0.5345 | 0.9472 | 0.9221 |
| **Recall** | 0.6132 | 0.6113 | 0.5924 | 0.6132 | 0.6094 | 0.5584 |

**Table 5.6:** Clustering detections after adaptive road scanning and an imprecise classifier ($t = 0.5$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.1709 | 0.2684 | 0.2669 | 0.3298 | 0.4250 | 0.2697 |
| **Recall** | 0.9471 | 0.8584 | 0.7285 | 0.9396 | 0.8339 | 0.7226 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.5879 | 0.6733 | 0.4160 | 0.8211 | 0.8126 | 0.7297 |
| **Recall** | 0.9396 | 0.7622 | 0.8226 | 0.9358 | 0.7528 | 0.6113 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.6945 | 0.8695 | 0.8157 | 0.0583 | 0.8426 | 0.7895 |
| **Recall** | 0.9396 | 0.9056 | 0.8188 | 0.9509 | 0.8792 | 0.7716 |

**Table 5.7:** Clustering detections after probabilistic 3D scanning and a precise classifier ($t = 0.85$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9301 | 0.8973 | 0.8457 | 0.9278 | 0.8904 | 0.8469 |
| **Recall** | 0.8792 | 0.7584 | 0.6622 | 0.8735 | 0.7358 | 0.6471 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.9275 | 0.8850 | 0.8147 | 0.9275 | 0.8775 | 0.8139 |
| **Recall** | 0.8698 | 0.7264 | 0.6056 | 0.8698 | 0.7169 | 0.5943 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.9168 | 0.9274 | 0.9272 | 0.3959 | 0.7371 | 0.8884 |
| **Recall** | 0.8943 | 0.8924 | 0.8660 | 0.8943 | 0.8943 | 0.8860 |

**Table 5.8:** Clustering detections after probabilistic 3D scanning and an imprecise classifier ($t = 0.5$).

| **Mean shift** | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma_s$ | log(1.3) | | | log(1.6) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.3661 | 0.4335 | 0.4375 | 0.5849 | 0.6274 | 0.6315 |
| **Recall** | 0.9962 | 0.9547 | 0.8396 | 0.9943 | 0.8962 | 0.7924 |
| $\sigma_s$ | log(2.4) | | | log(3.2) | | |
| $(\sigma_x, \sigma_y)$ | (4,8) | (8,16) | (16,32) | (4,8) | (8,16) | (16,32) |
| **Precision** | 0.8429 | 0.8549 | 0.8043 | 0.8456 | 0.8551 | 0.8108 |
| **Recall** | 0.9924 | 0.8339 | 0.6981 | 0.9924 | 0.8245 | 0.6792 |

| **Accumulative** | | | | | | |
|---|---|---|---|---|---|---|
| | overlapping | | | xys | | |
| threshold | 0.25 | 0.5 | 0.75 | 1 | 5 | 10 |
| **Precision** | 0.2501 | 0.8361 | 0.8156 | 0.0130 | 0.0980 | 0.4470 |
| **Recall** | 0.9962 | 0.9433 | 0.8264 | 0.9962 | 0.9867 | 0.9396 |

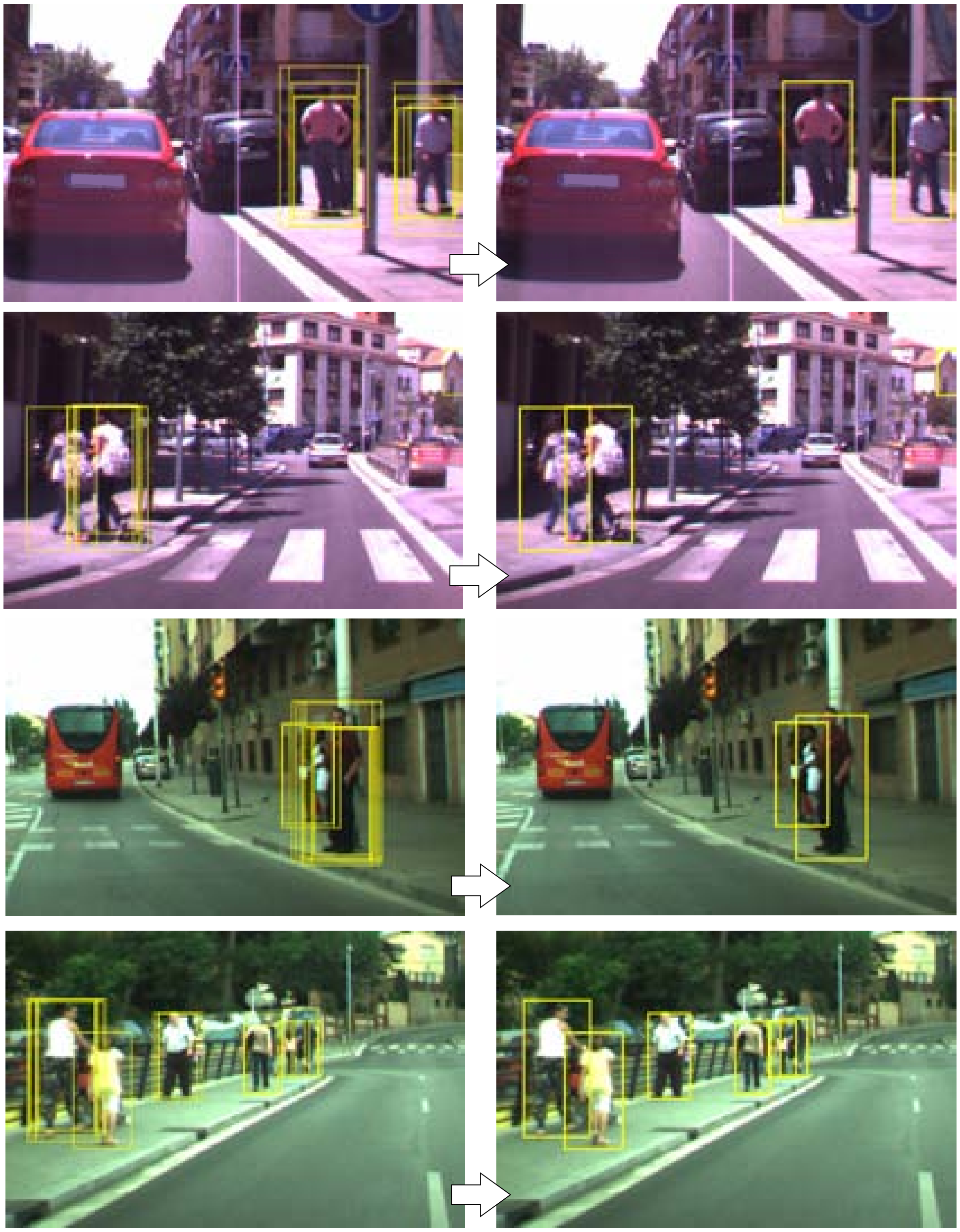


**Figure 5.6:** Results after using accumulative clustering on the detections provided by probabilistic 3D and SHOG + SVM.

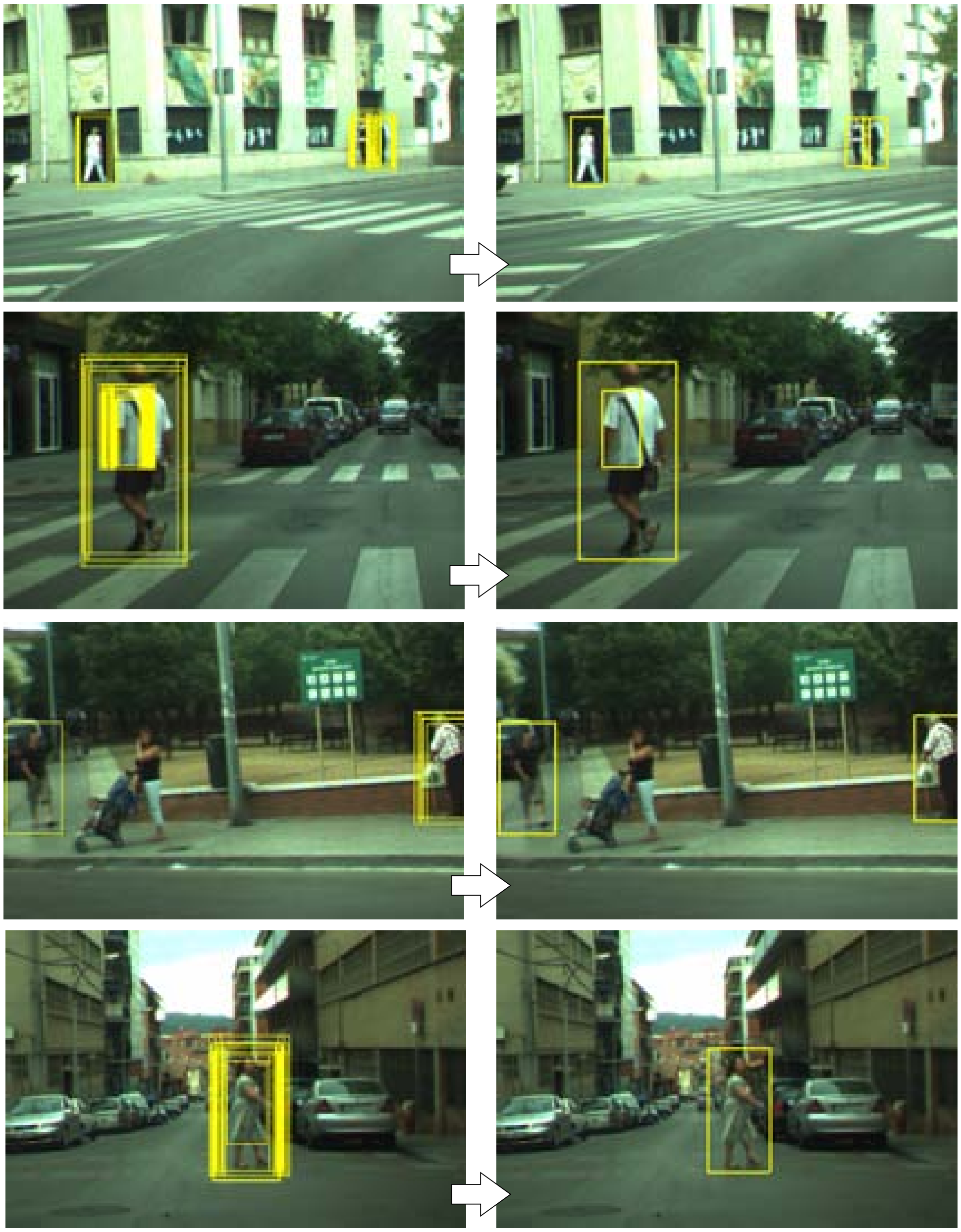


**Figure 5.7:** Results after using accumulative clustering on the detections provided by adaptive road scanning and SHOG + SVM.

## 5.2 Shape extraction via non-explicit shape models

As has been seen along the thesis, pedestrian detection mainly works with windows, both as interesting regions to feed the classifier and as the output of the system itself. It is clear that windows provide in many case sufficient information for the algorithms, however, the refinement of the window into a pedestrian shape, even in an approximate fashion, has many potential benefits:

- the tracking module can get rid of the background regions in a detection window to construct a more refined model (e.g., color, shape, etc.) than using raw boxes;
- the module can perform a verification-after-refinement (not necessarily replacing an hypothetic first verification) by making use of the pedestrian silhouette;
- although there are not well-established application module algorithms, the knowledge of the target's shape poses as an attractive intuitive cue for the driver in any visualization device, for example either on a separate screen or on the windshield.

In Chap. 2 two main proposals capable of performing the extraction of pedestrians shape are described: [67] and [92]. In both proposals, i.e., the Chamfer system and implicit shape model with masks, an annotation of the pedestrians shape is required. Almost every researcher in object classification is aware of the problems of manual annotation, i.e., it is a boring and tiresome task. Moreover, if one talks of explicit shape annotation, the task turns into an even more tedious and slow job which subject to imprecisions as a result of human errors.

In this thesis we have researched the possibility of performing shape refinement without explicitly annotating pedestrian silhouettes for training. This work has been published in the Master Thesis recently presented by Hilda Caballero [32].

### 5.2.1 Basic algorithm

The proposed algorithm makes use of two basic ingredients: segments extraction / description and bag of words representation. Figure 5.8 illustrates how the model training works. The algorithm can be summarized in the following steps:

a. First, image edges are extracted by making use of Canny algorithm with automatic thresholds.

b. Segments Extraction: segments are extracted from the resulting edge image by the use of a modified version of Connected Component Labeling (CCL) algorithm, which we name Gradient-CCL. The original CCL finds all connected pixels in an image attending to a neighborhood criterion and assigns a unique label to all pixels in the same component. In our proposal, we make use of both orientation and spatial neighborhood to compute the components, that is, a pixel $p$ is linked to a component $C$ only if $p$ is not null in the Canny edges image, it is at one pixel distance from $C$ and its gradient orientation is similar to the mean of the pixels in $C$.

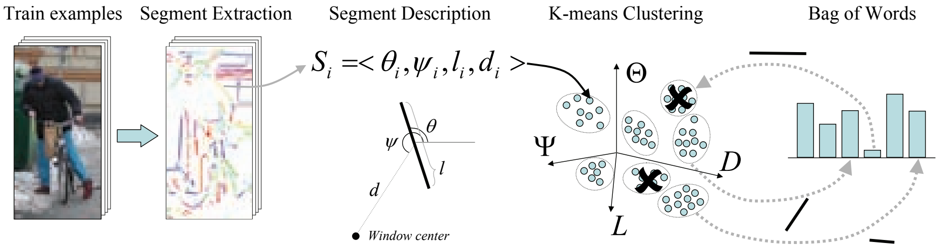


**Figure 5.8:** Training steps of the shape extraction algorithm.

c. Segments Description: Each connected component resulting is then represented in a line by joining its ending points. This dimensionality reduction of the original edge segment is aimed at easing the computation of the descriptor without the danger of losing information, given that the edge components are typically straight lines thanks to the CCL-Gradient algorithm. Then, line each segment is described by four measures: its mean gradient orientation angle $\theta$, its length $l$, the relative angle between the center of the segment and the center of the window $\psi$ and the distance from the center of the segment to the center of the window.

d. Segments Clustering: By making use of the descriptor vector of each segment, a K-means clustering is performed in the segments space, getting compact representations of the typical segments found in pedestrian windows. In order to perform a correct clustering, the axes are normalized between 0 and 1 by using the necessary transformations to the previously mentioned measures.

e. Bag of Words Representation: The final bag of words model is computed by constructing an histogram of segments (the model $M$) in which each cluster mean represents a bin and the number of occurrences inside each cluster represent the score of the bin. Since the most usual segments (i.e., legs or shoulders segments) will have high scores in the histogram and segments coming from background clutter will have low ones, lowest ones are filtered out in order to discard not representative segments in the pedestrian model.

The model can then be used to extract the pedestrian shapes from the test windows. Figure 5.9 illustrates the process, which can also be split in different steps:

a. Segment extraction and description by following the same procedure as in the training.

b. Match each line segment in the test image to the clusters in $M$. If not matched, then discard it since it belongs to background.

c. Recover original edge segments from the remaining segments.

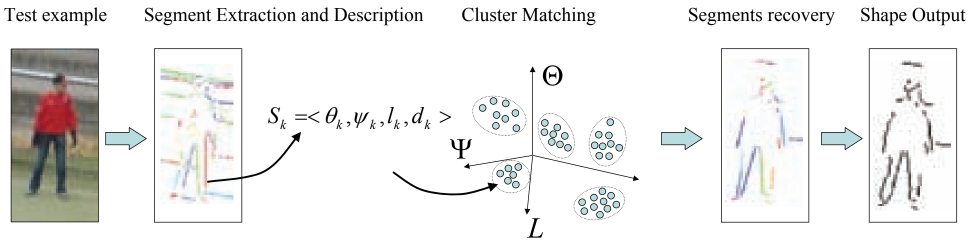

Test example
Segment Extraction and Description
Cluster Matching
Segments recovery
Shape Output
$S_k = <\theta_k, \psi_k, l_k, d_k>$
Θ
Ψ
L

**Figure 5.9:** Testing steps of the shape extraction algorithm.

### 5.2.2 Additional Improvements

Once the basic algorithm has been developed and validated, we propose two additional steps that improve the experimental results. Both them are performed in the training phase:

- We realized that many irrelevant segments coming from both background clutter and pedestrians clothes are integrated in the model. These segments came from edges in shadowed areas belonging to the same object (e.g., trousers or grass), but did not belong to any of the major edges that we aimed to find in the windows. In order to reduce this clutter we applied a preprocessing using Bilateral Filtering algorithm just before the Canny edge finding. Bilateral Filtering, proposed in 1998 by Tomasi and Manduchi, smooths the image while preserving edges by means of a nonlinear combination of nearby image values. The method combines gray levels or colors based on both their geometric closeness and their photometric similarity, and prefers near values to distant values in both domain and range.

- The second improvement is focused on eliminating segments in the model coming from common background structures such as sidewalks or the horizon line. This has been achieved by also taking into account background windows when training. Similarly to the pedestrian windows, segments are extracted and described, but in this case they are accumulated in a histogram to compute a bag of words model of the background. This histogram is then subtracted from the pedestrians' $M$ in order to filter out the high scoring segments that do not correspond to the pedestrians silhouette.

### 5.2.3 Experimental results

The reason to use INRIA instead of our proposed CVC-02 is that we preferred to present the results with images of a higher resolution than the typical ADAS ones of CVC-02. In addition, since this technique is very likely to be used in other areas in which INRIA is the de facto dataset, we think that it is convenient to first study the behavior of the algorithm on it. However, samples are $64 \times 128$ pixels, so the proposed algorithm is likely to be applicable to pedestrians from 0 to 20-25 m with a typical ADAS camera, which is left for future research. The sample windows have been divided into three sets: training, which contains 519 images and their mirrors from INRIA-train, used to train the model; validation, which contains 25 samples and their mirrors from INRIA-train, used to tune the parameters and the performance improvement of the aforementioned additional steps; and test, which contains 100 images and their mirrors taken from INRIA-test, which is used to compute the final experiments. Both validation and testing samples are annotated by manually defining the silhouette of the pedestrian. Additionally, a background set with 4 152 images is used to train the background histogram previously mentioned.

The measure used to compute the error is the Symmetric Chamfer Distance (see App. D), and the baseline used to assess whether the algorithm provides significant results is the Canny edge image. Four different configurations of the method are used:

- **Configuration A**: Euclidean distance used in the bag of words creation and in the segments matching step.
- **Configuration B**: Mahalanobis distance used in the shape retrieval step to decide whether a given segment belongs or not to a cluster.
- **Configuration C**: Background model is added to the computation of the model.
- **Configuration C+**: Bilateral filtering.

Table 5.2.3 shows the mean and standard deviation of the error. As can be seen, all tested configurations give smaller error than the baseline, and the best configuration is D, which features Mahalanobis distance, a background model and bilateral filtering[2].

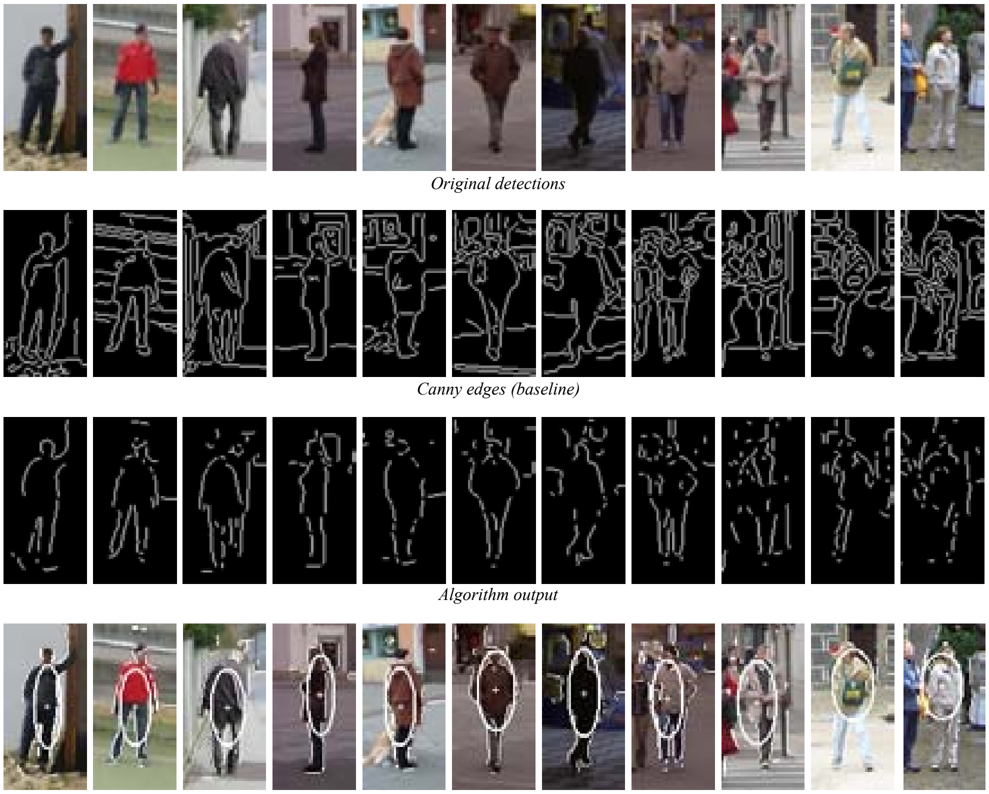
**Table 5.9:** Error mean and standard deviation of the different shape extraction algorithm configurations.

| **Baseline** | |
|---|---|
| **mean** | **stdev** |
| 8.8698 | 1.3661 |

| **Pedestrian Models Configurations** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **A** | | **B** | | **C** | | **C+** | |
| **mean** | **stdev** | **mean** | **stdev** | **mean** | **stdev** | **mean** | **stdev** |
| 7.6079 | 1.6671 | 7.8719 | 1.7256 | 7.1826 | 1.8611 | **6.8943** | **1.9082** |

Finally, we present some qualitative results in Fig. 5.10. The first eight columns show satisfactory results in which the major lines of the pedestrian have been selected and background clutter has been significantly removed. As can be seen, the resulting segmentation provides a rough silhouette of pedestrians that can be used for the aforementioned tasks. The last three columns represent incorrect retrieved shapes by the proposed algorithm. We have found that the errors come mainly from the following sources: non-existing edges as a result of similar pedestrian-background color (e.g., similar trousers and ground color); noisy clutter that is incorrectly added to the bag of words model, but not avoidable by the background model (e.g., varied small segments mainly on the top area or clothes texture).

[2] We have demonstrated the statistical significance of these errors compared to the baseline by the use of a one-way analysis of the variance (ANOVA).



Original detections

Canny edges (baseline)

Algorithm output

Overlayed output and center of mass

**Figure 5.10:** Experimental results of the proposed algorithm.

### 5.2.4 Discussion

As has been demonstrated, the proposed method provides promising results in the task of shape refinement without requiring annotated silhouettes for training. It can also be seen in the qualitative results that verification or tracking algorithms can clearly take advantage of the refinement output. For instance, the segments center of mass and the ellipses shown in the last row of Fig. 5.10 are potential cues to compute color models of detected pedestrians useful for tracking.

However, this work has been basically exploratory, so we consider it as a strong line for future work aimed at resolving some important questions:

- Once having stated that shape retrieval does not necessarily involve the so time consuming annotations, it is mandatory to compare the proposed algorithm performance to the state of the art annotation-based methods described previously. This task is not easy given the lack of publicly available datasets and benchmarks, similarly to what happens in many other problems in pedestrian detection.

- In our experiments we have made use standard $64 \times 128$ pixels windows, which are likely to be directly used in a pyramid-like foreground segmentation approach. In order to adapt the proposed algorithm to any possible size of window, some adjustments would have to be made in terms of segment normalization, concretely in the distance and length measurements of the descriptor. Besides, a further study is needed to evaluate the performance of the algorithm according to the distance (i.e., size) of the target.

- Similarly to a classifier, a multi-pose algorithm would improve the performance of the method.

- Finally, the algorithm is subject to many improvements in its components, for example, different descriptors, different distance measures, clustering algorithms, etc., aimed at resolving the highlighted main problems in the experimental results.

## 5.3 Summary

Detections refinement for PPSs is another been poorly investigated task, in the literature just a few papers present any kind of research in this area. This chapter addresses two relevant topics: detections clustering and silhouette extraction. In the former we have formalized one of the few existing algorithms in the literature and also our proposed one. Next we have studied these algorithms along with the effects of foreground segmentation and classifier precision on them. The experimental results provide insights that set important conclusions in this topic, for instance, the performance of clustering algorithms tends to be similar so the criterion to select them shall be based on for example computation time. Regarding the latter topic, a novel shape extraction that does not require explicit training shape annotation has been presented. In spite of the early stage of this research the results are promising and seem applicable for pedestrians in the high-risk region of our system, that is, from 0 to 25 m.

*Tokyo diary*
Barbara Kirsch, oil on canvas, 1996
(Private Collection)

Similar to the painting illustrating the previous chapter, there is a direct relation between *Tokyo diary* and the current chapter. First, it represents the view from the inside of a vehicle, which is similar to the driver assistance systems imagery. Second, the painting uses a grayscale palette, which is the common used in pedestrian detection given the few information that color provides to the pedestrian model. And finally, the feeling of motion, changing and not clearly defined shapes that make Computer Vision a tough problem in general. From an even more personal viewpoint, the light at the end of the tunnel inspires us that the end of this work is not far and that, although there is still a long way to engineer a pedestrian protection system robust enough to be commercialized, each day that passes there is a clearer path to follow.

# Chapter 6

# Pedestrian detection system

In order to complete the thesis, in this chapter we present our proposal of a complete pedestrian protection system. As usual, in Sect. 6.1 we present the evaluation methodology used throughout the chapter. In Sect. 6.2 we describe the components of the system, and in Sect. 6.3 both quantitative and qualitative experimental results are presented. Finally, Sect. 6.4 draws the discussion.

## 6.1 Evaluation methodology

The final system is evaluated on 15 urban video sequences, containing 4 364 frames with 7 983 pedestrians (for more information refer to the details of CVC-02-System in App. B). The training of the different modules has been done using the corresponding datasets, not overlapped with the testing set, that is, to train the classifier we use the CVC-02-Classification subset.

Regarding the evaluation protocol we plot image-based FPPI curves for each sequence. The reader shall keep in mind that since the system does not contain a tracking module, we do not evaluate tracks but frame-based detections.

## 6.2 Proposed system

The system is composed by several of the proposed algorithms in the previous chapters, namely:

- Foreground Segmentation. Despite that probabilistic 3D scanning is the most promising approach, for the moment we opt for **adaptive road scanning** as the foreground segmentation algorithm attending to the large dataset and the computational resources available. In addition, we will perform the **3D filter-**

**ing** just in the near range pedestrians. However, as soon as the implementation of probabilistic 3D approaches realtime it will be the preferred algorithm for the system.

- Object Classification. **SHOG+SVM** has provided the best performance, so it is the chosen option for this module.
- Refinement. For the clustering we use **accumulative algorithm** with the best parameters according to the study: overlapping with 0.5 threshold.

We have not included the silhouette extraction algorithm in the system since the testing with our dataset has been left for future work. In addition, we have not included tracking as it is out of the scope of this thesis.

## 6.3 Experimental results

In this section we first provide quantitative evaluation by means of performance statistics for each individual sequence and then we also provide qualitative data by including several image results that include most of the representative hit and fail cases.

We have performed an individual image-based evaluation for each test sequence. As can be seen in Fig. 6.1, the performance of the system in the different sequences greatly varies, and the DR ranges from 0.4 to 0.75, in average 0.5425±0.17, when FPPI= $10^0$ is imposed.

Figures 6.2 and 6.3 illustrate the typical results obtained when fixing the thresholds to get 1 FPPI. We have tried to show the most representative frames so the discussion and conclusions are profitable. Figure 6.5 illustrates eight consecutive frames from sequence 8 in which the vehicle is driven over a speed bump and the adaptive road scanning successfully deals with the scene. Figures 6.6 and 6.7 plot the distance from the camera to each detection, computed as the mean of the depth pixels inside the window by making use of the 3D reconstruction. Finally, in Fig. 6.4 we present representative snapshots from all the sequences.

## 6.4 Discussion

From the evaluation plots shown in Fig. 6.1 three types of performance can be distinguished depending on the difficulty of the sequence, which varies according to factors like road slope, clutter, etc. (see App. B). Hence, it can be said that the system performs well in sequences 9, 10, 11 and 12; average results are found in 2, 4, 13, 14 and 14; and it is susceptible of improvement in sequences 1, 3, 6, 7 and 8, for example by using tracking. In fact, it is hard for the system to deal with sequences with big road slope variations, which is one of the key points to keep researching in the future. Clutter and crowded sequences are also a challenge: by having a look at the properties of the datasets one realizes that the poorest results correspond to crowded scenarios. In fact, they pose two problems: first, clustering in crowds is more difficult in the sense that perfect candidates classification can lead to false negatives if two or more pedestrians are very close to each other; second, since the pedestrians are mostly moving, the variations between the foreground and the background are increased, so more FPs appear.

As can be seen, for this threshold there exist false positives in almost all the frames. There are also some false negatives, mainly as a result of a wrong road estimation. However, one important point that is clearly noticed is that both false positives and false negatives are mostly spurious. For example, the two sequences in the latter figure show a man and a woman that are perfectly followed along all the sequence, whereas false positives appear in isolated frames, which is an important aspect to take into account. It is also worth to mention that given the intermittent nature of misdetections, specially false positives, the system is very likely to improve by attaching a tracking stage. By the use of temporal coherence that, for instance, just highlighted detections present in three consecutive frames, the results would be competent enough. In addition, although in our experiments we have avoided the use of probabilistic 3D scanning given the current computational cost, it is also likely to solve the problem of misdetections when the road is wrongly estimated according to the quantitative results of Chap. 4, specially for far distance pedestrians.

Finally, we shall highlight that from the beginning we tried to avoid the typical easy sequences with isolated pedestrians that can be found in the literature, so getting performance rates of around 50% with 1 FPPI is a good performance, indeed comparable to the state of the art, which makes us be optimistic of the results to be achieved in near future research.

## 6.5 Summary

In this chapter we have presented the performance results, both quantitative and qualitative, of a proposed pedestrian detection system that makes use of several components presented throughout the thesis. The results let us be optimistic in the sense that, although the performance rates are still below the requirements of serial commercialization, the nature of misdetections, intermittent and mostly caused by wrong road estimations, does not seem an insurmountable problem.

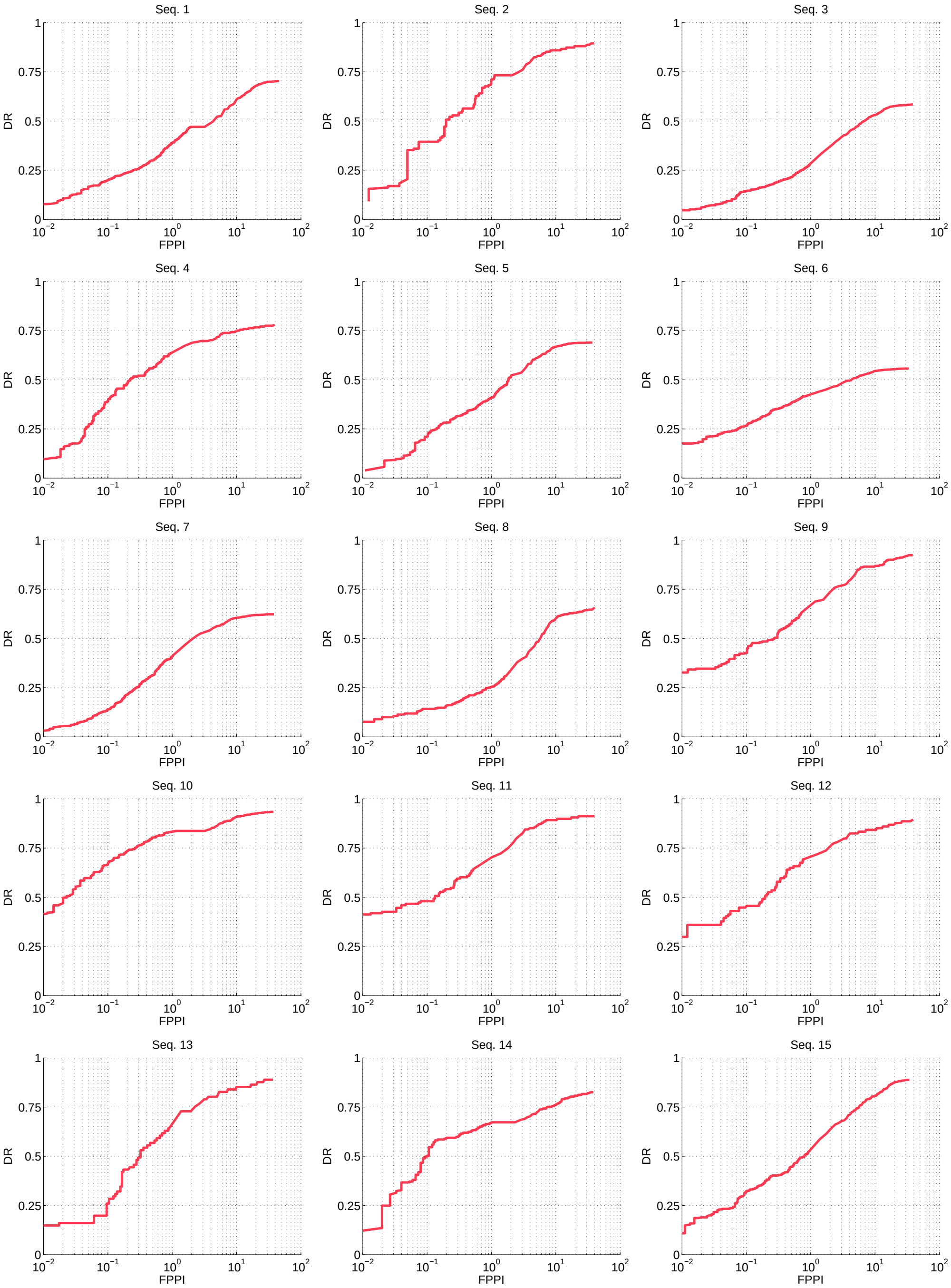


**Figure 6.1:** Image-based evaluation of the proposed system for each test sequence.

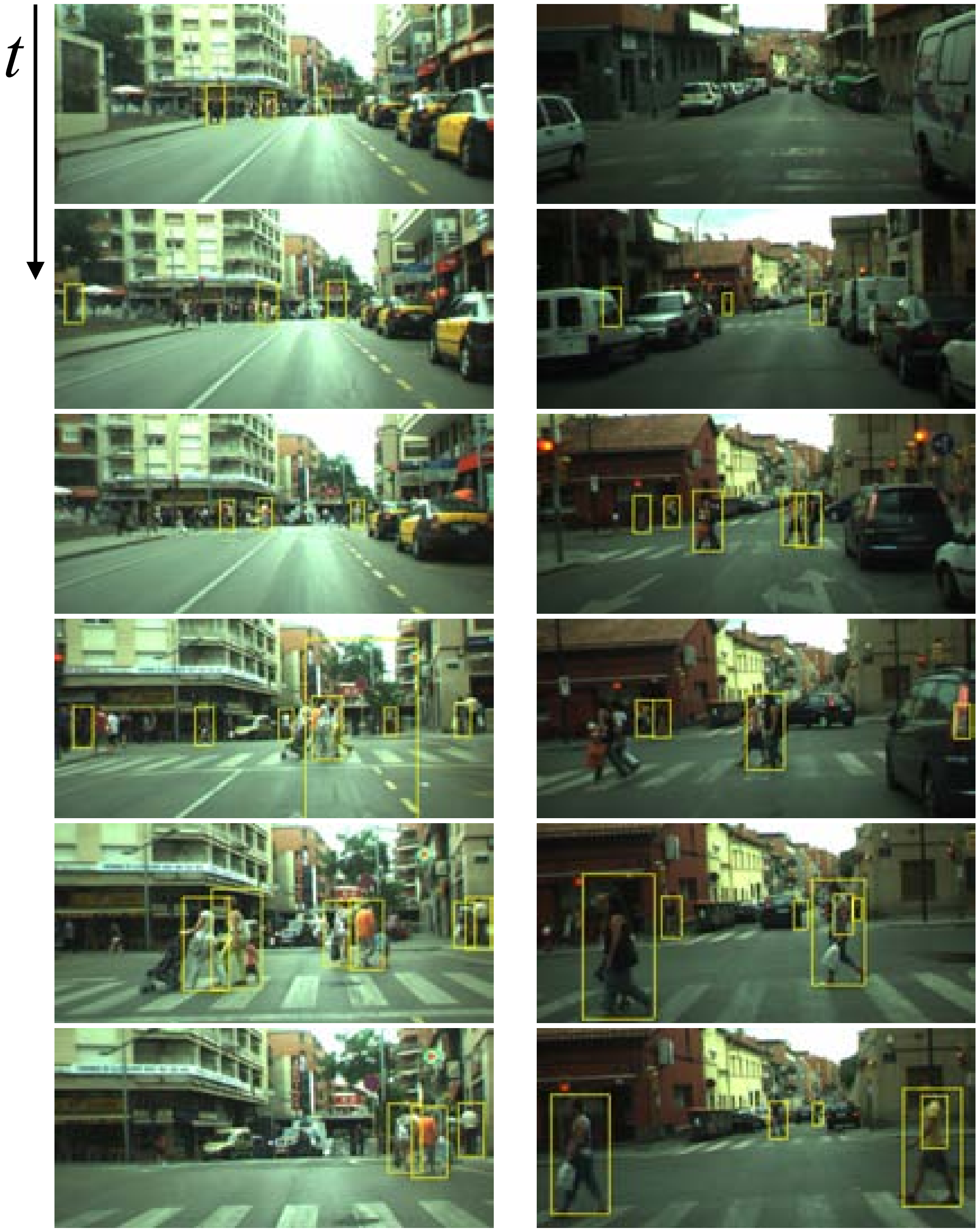


**Figure 6.2:** Representative results when FPPI= $10^0$ for sequences 6 and 7.

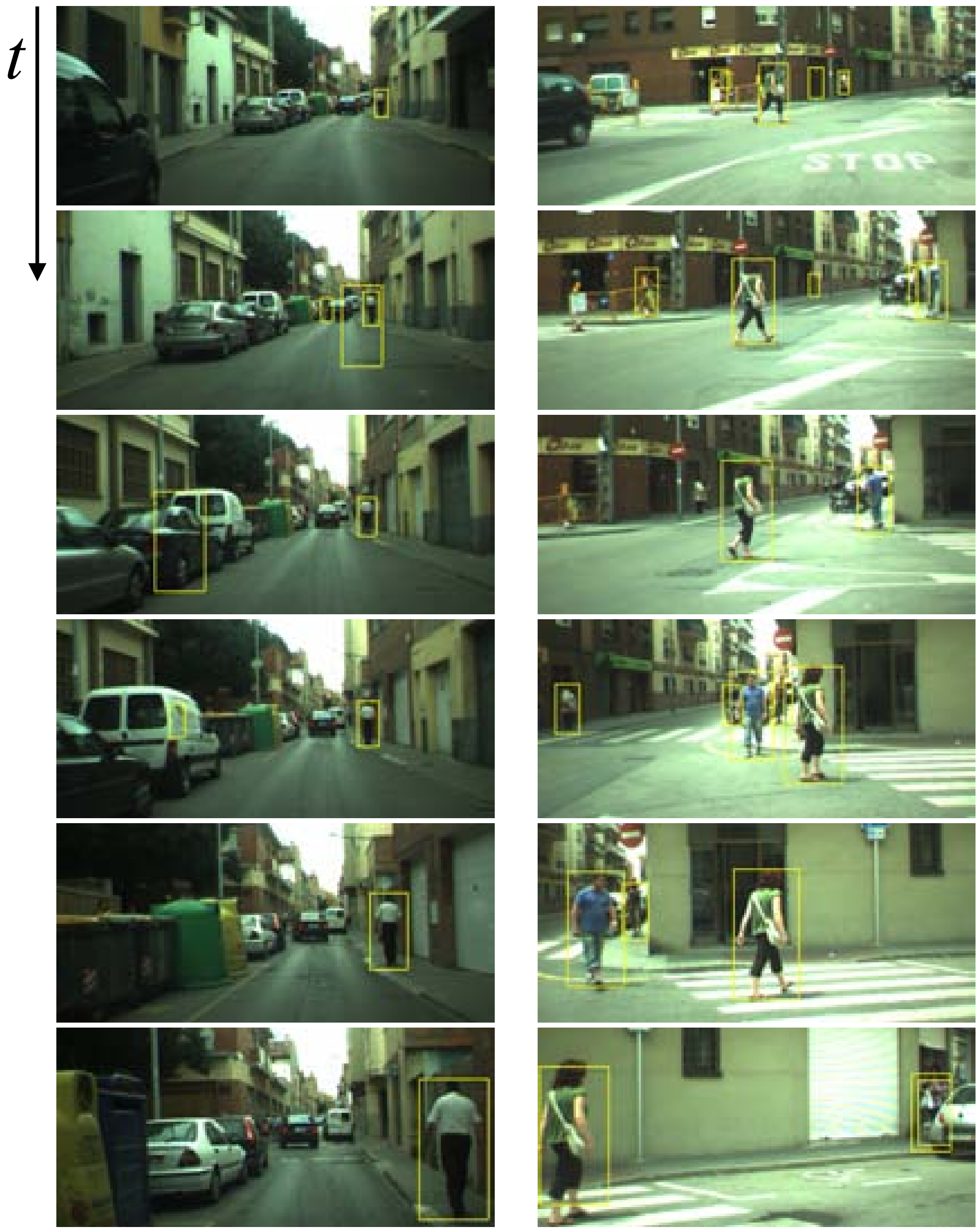
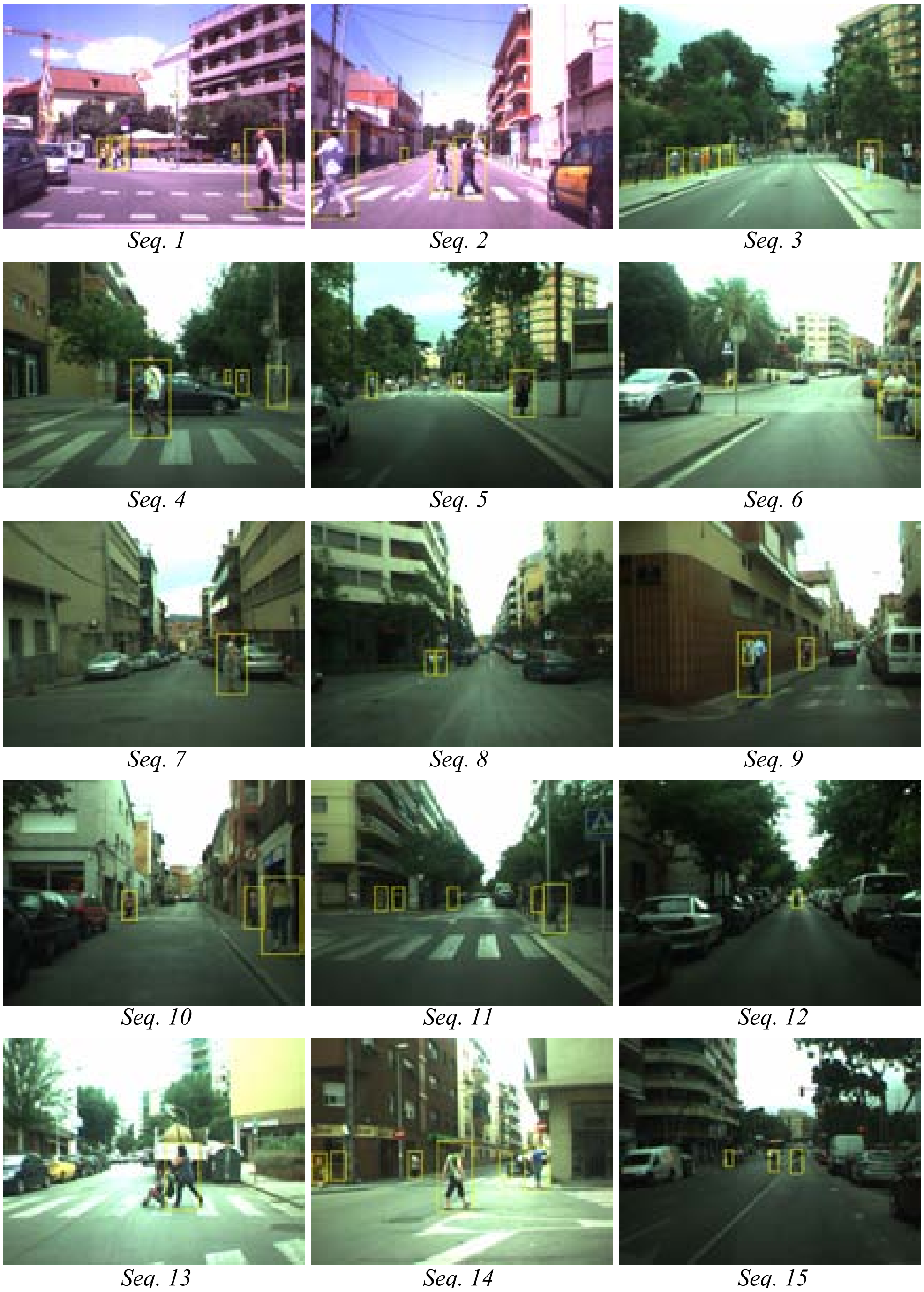


**Figure 6.3:** Representative results when FPPI= $10^0$ for sequences 9 and 14.



*Seq. 1* *Seq. 2* *Seq. 3*

*Seq. 4* *Seq. 5* *Seq. 6*

*Seq. 7* *Seq. 8* *Seq. 9*

*Seq. 10* *Seq. 11* *Seq. 12*

*Seq. 13* *Seq. 14* *Seq. 15*

**Figure 6.4:** Representative results snapshots when FPPI= $10^0$ for all sequences.

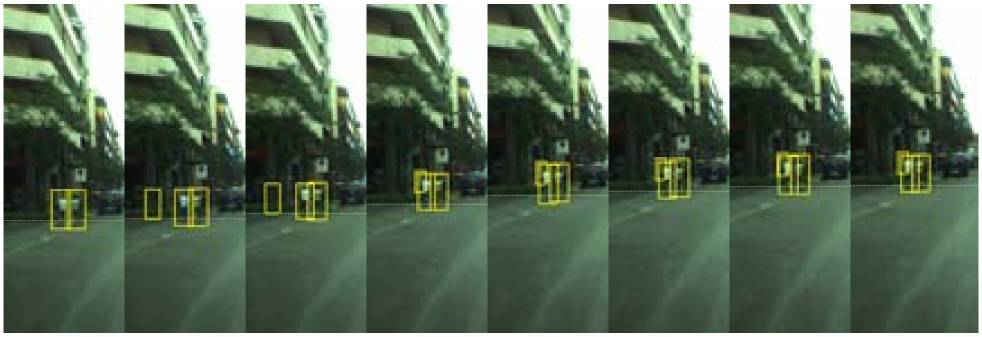


**Figure 6.5:** System results when passing through a speed bump.

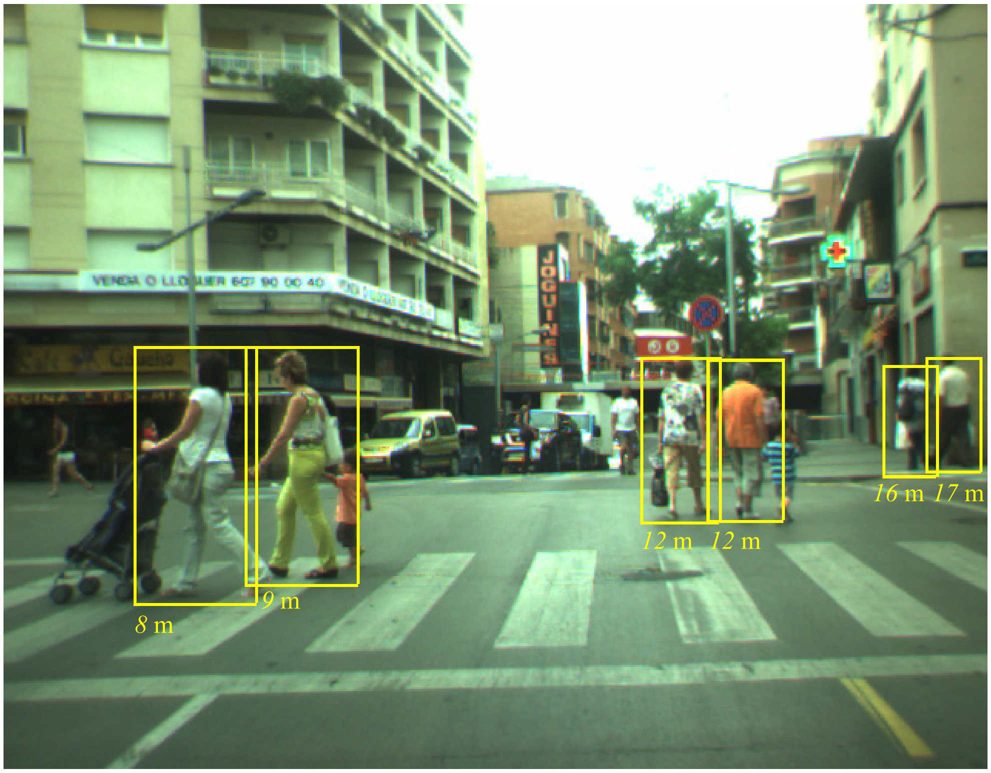


**Figure 6.6:** System results with the distance of each detection.

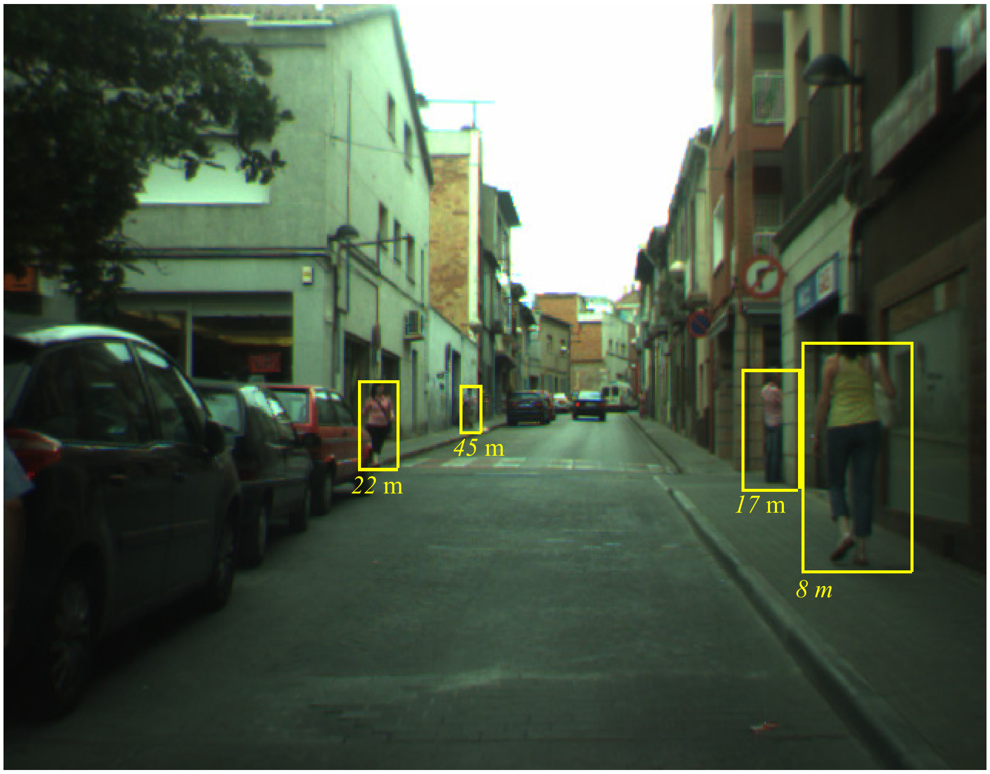


**Figure 6.7:** System results with the distance of each detection.

# Chapter 7

# Conclusions

Pedestrian protection systems represent a key technology to reduce the number of accidents between pedestrians and vehicles. Given the difficulties that such systems shall overcome, that is, realtime detection of changing targets in uncontrolled outdoor scenarios, pedestrian protection is by no means an easy task. From our point of view, research was too focused on specific tasks of the system like classification and forgot the relation between them. In this thesis we have developed the research from a global viewpoint.

## 7.1 Summary and contributions

Although each chapter contains a specific discussion section that analyses and points out the most relevant advantages and disadvantages of the explored algorithms and of their combination between the modules of the proposed architecture, in this chapter we highlight some more global conclusions, which are in fact linked with the contribution of the thesis.

**In the survey of the state of the art, we have extensively reviewed the literature by first introducing a general architecture that consists of different modules**, each with its own objectives, in which we fit every analyzed technique in the literature. As has been seen, this general architecture is of crucial importance to analyze the literature in a sensible and ordered way. In the chapter we have highlighted a set of interesting points that lead the thesis studies and in fact will lead the future lines of research, like problems usually omitted in the literature (e.g., foreground segmentation) or techniques that have demonstrated their classification capabilities (e.g., histograms of oriented gradients). According to the review, it can be said that there is a clear research trend in every module. For example, the promising algorithms in foreground segmentation are the road based ones; the research in classification is mostly focused on gradient-based features and several typical learning algorithms, but recent multiclass/multipart approaches are also gaining importance;

and the Kalman filter is the most used algorithm for the tracking module.

In the survey we have also highlighted the lack of datasets. In order to evaluate all the different proposals, **we have introduced a pioneer multi-purpose dataset aimed at being utilized as an evaluation framework for different modules of a PPS**: foreground segmentation, classification and whole system.

Foreground segmentation has traditionally been a poorly researched component of the system. **A novel performance evaluation methodology has been set for candidate generation algorithms by proposing new protocols and measurements**. In addition, **we have formalized the existing algorithms and proposed new ones**. In fact, the potential of the proposed candidate generation algorithms has been demonstrated, and adaptive road scanning and probabilistic 3D scanning are the techniques to be used in this module. From our point of view, the presence of the stereo information for this task is mandatory for any future system. Without this cue, the foreground segmentation is restricted to a blind scan in which the only investigation to be done is to choose the number of candidates the system can manage.

**We have analyzed the classification in the context of the system, i.e., relating it to the foreground segmentation, instead of trying to improve the classifier as an independent entity in the system**. In this way, future improvements in classification will be able to benefit from the study, and will likely fit the progress presented in this thesis. As was seen, although HaarEOH+AdaBoost provide quite confident performance at low computational requirements, they are outperformed by the SHOG+SVM classifier, which in fact is used for the system experiments. It has been demonstrated that the so-called HOG are not always the best option for human detection, as SHOG outperform it in the ADAS dataset. In addition, the performance of the classifiers when used in the context of a system greatly depend on the previous stages. For example, there are differences in performance between the studied classifiers depending on the foreground segmentation algorithm used.

**We have formalized two clustering algorithms and have demonstrated that their performance is quite similar, provided that their corresponding parameters are tunned correctly. Accordingly, we bet for the computation time criterion in order to select the algorithms for this task**. In addition, we have stated that the performance improvement when attaching clustering with respect to the raw candidate classifications is of about a 15% when using adaptive road scanning and a 5% when using probabilistic 3D scanning. In the context of detections refinement, we have also proposed a technique to go one step further than the typical boxes by avoiding to annotate shapes. **We have demonstrated the potential of the proposed shape extraction algorithm with a generic human dataset**, that is, INRIA, but the results let us think that the method is also promising for ADAS imagery.

Finally, **three techniques analyzed along the thesis have been integrated in a pedestrian detection system and tested in real urban scenarios**. The results show that although there exist misdetections, given the false positive rate imposed (1 per frame), they tend to be intermittent and are likely to be discarded by a future tracking step. In addition, the true positive detections are continuous in time, specially the ones occupying the high risk area.

## 7.2 Perspectives

Driver assistance systems, and particularly pedestrian protection systems, are a very young area of research. Hence, the future research possibilities are so numerous and diverse that they can easily occupy a chapter on its own. We condense the lines we consider of key importance in a few general points.

**Short term challenges.** The pursuit of a perfect PPS based on Computer Vision only shall be considered a long term goal. The development of a PPS that works under restricted conditions is already useful. For instance, a system that works only at daytime, under good weather conditions (no heavy rain/snow/fog), over a range of distances up to 50 m is, from our viewpoint, the first intermediate challenge for the community. According to [4], these conditions represent a very relevant scenario in accidents.

**Long term goals: focus on the real problem.** It is clear that many new proposals are tested on too easy data. Developing systems capable of working under restricted conditions is different from developing techniques that just work on high-resolution near non-occluded pedestrians, because this can lead to a loss of perspective of the real problem. Although it can seem strange to provide statistics of 10% DR or 10 FPPI, specially in front of other more traditional areas like face detection or object classification, this *poor* performance in realistic complex examples is more useful for the community than presenting a 99% in nearly toy examples.

**Face the problem globally.** In addition to developing the individual parts of a PPS, which is one of the keys to reach good performance rates, a global view of the problem can lead us to interesting conclusions as the ones assessed in this thesis.

**Overall vision of future ADAS.** When talking about a global viewpoint, one also has to have in mind that a PPS is likely to work with many other ADAS. This leads to a point in which the different systems have to share sensors and computation time, which in fact has both disadvantages such as the restrictions when choosing sensors, but also advantages in the sense that techniques like stereo reconstruction, free space analysis and even sensor fusion data can be shared between them.

# Appendix A

# Acquisition system

This appendix details the acquisition system employed to record the sequences used for the datasets generation and final system evaluation. The system consists of a camera, attached to the vehicle windshield via two suction pads (Fig. A.2($a$)), and a laptop, connected via IEEE-1394 port. The camera is a Point Grey Bumblebee, which consists of two Sony ICX084 color CCD sensors with the following parameters:

- Focal length: $6mm$.
- Effective Field of View: 43.07° HFOV and 32.97° VFOV. (see Fig. A.1)
- Resolution: $640 \times 480$ pixels.
- Effective sensor: 4.736(H)$\times$3.552(V)$mm$.
- Stereo baseline: $12mm$.
- Bayer pattern RGGB, demosaiced using the camera software.
- Frame rate of about 10fps, saving images to a 7 300$rpm$ internal laptop HDD.

The system has been mounted on two different sedans, a *Mitshubishi Colt* and a *Ford Escort Mk5b*, thus both height and pose of the camera varies due to the different host vehicle dimensions and windshield positions. Figure A.2 shows some photographs of the system components.

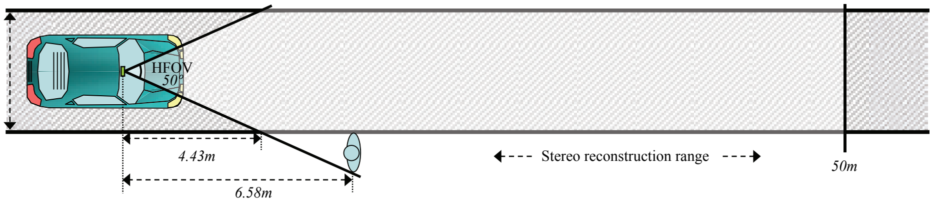


**Figure A.1:** Acquisition range of our system, which fulfills the system requirements described in Chap. 1.

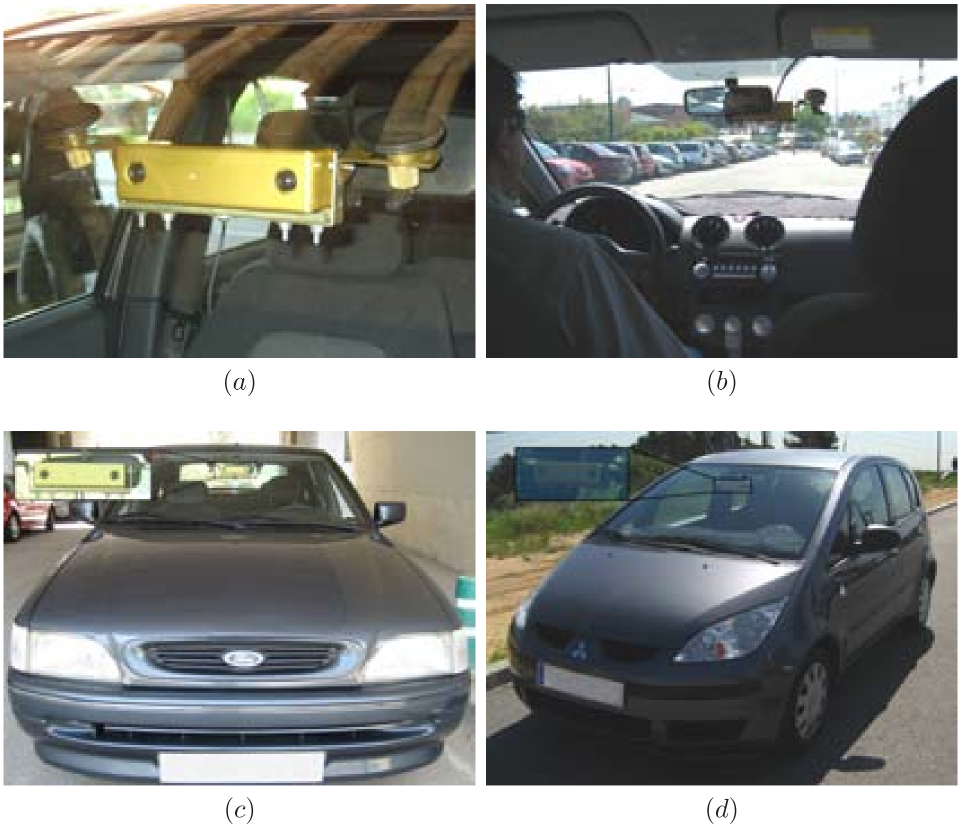


**Figure A.2:** Acquisition system. (*a*) Camera. (*b*) View from inside of the vehicle. (*c*, *d*) The two acquisition vehicles used.

# Appendix B

# CVC-02 Pedestrian Dataset

In this thesis we present a dataset divided in three subsets, each focused on a different task of pedestrian protection systems. In this appendix we summarize their contents and provide details that were ommited in the corresponding chapters.

The imagery has been recorded in different acquisition days during 2008 and 2009 by mounting the system on two different sedans (see App. A). The recording scenes are urban scenarios around Barcelona.

The recorded sequences have a total length of 3 5 hours, which represent about 130 000 color stereo frames at 10fps. Frames size is $640 \times 480$ pixels. When extracting the corresponding datasets we carefully assure the independence between training and testing sets by using different acquisition days and even acquisition locations. The annotation has been carried out by the author by making use of a self-made program in Matlab, also made available together with this thesis. It consists in $< x, y, w, h >$ vectors representing each pedestrian, where $x$ and $y$ are the center coordinates, and a string tag stating the label of the pedestrian (either *PEDESTRIAN-OBLIGATORY* or *PEDESTRIAN-OPTIONAL*) when needed. All the annotations of a frame are stored in an individual .txt file.

When a label is included, it states whether the annotation shall be treated as obligatory or optional (which can be used to generate different performance curves) according to the following criteria:

- Obligatory: Pedestrians between 0 and 50 m (i.e., bigger than 24 pixels high in our system) on the ground plane or pavement. Partially occluded pedestrians by self-carried objects (e.g., trolleys or prams) or young children are also obligatory.
- Optional (irrelevant to performance statistics):
  - Too young children to go alone in the street (these are supposed to be took by the hand by an adult).
  - Significantly occluded by an object or another pedestrian.
  - Partially out of the image.
- Not-annotated (counted as false positive in statistics):

- People riding bicycles or motorbikes.
- People out of the ground height which are not a target of PPSs (e.g., people in balcony).

The annotations have been made by selecting the top $y_1$ and bottom $y_2$ borders of the pedestrians, which define the height $h = y_2 - y_1$ and their center $y = (y_2 - y_1)/2$ coordinate in the $Y_I$ axis, and then fixing the center $x$ position in the $X_I$ axis. The width is automatically defined as $w = h/2$ to maintain a 1/2 aspect ratio. Notice that the annotations cover the pedestrian with no extra space, letting the later classifier select the appropriate margin.

The datasets are available in `http://www.cvc.uab.es/adas/datasets`, while the annotation program can be found in `http://www.cvc.uab.es/~dgeronimo`.

## B.1 Candidate Generation (CVC-02-CG)

CVC-02-CG is used for foreground segmentation evaluation (Chap. 3). It consists of 100 frames with 266 pedestrians annotated, the smallest one of $12 \times 24$ pixels and the biggest one of $156 \times 316$ pixels. We provide color, depth and 3D points of each frame. Figure B.1 illustrates the nature of the data provided in the dataset.

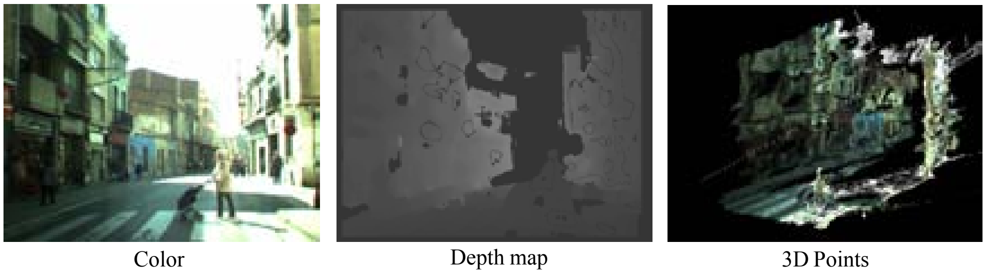


**Figure B.1:** Provided data for each frame in CVC-02-CG dataset.

## B.2 Classification (CVC-02-Classification)

CVC-02-Classification is used in the evaluation of classifiers and foreground segmentation approaches of Chap. 4 and the clustering in Chap. 5. All cropped images and frames are provided both in color and depth. Figure B.3 illustrates some examples of the dataset divided in three different size ranges.

- The **Training set** consists of:
  - 1016 cropped positives and their mirrors (original scale and $64 \times 128$).
  - 7650 cropped negatives using Sliding Window (original scale and $64 \times 128$).
  - 7650 cropped negatives using Adaptive Road Scanning (original scale and $64 \times 128$).

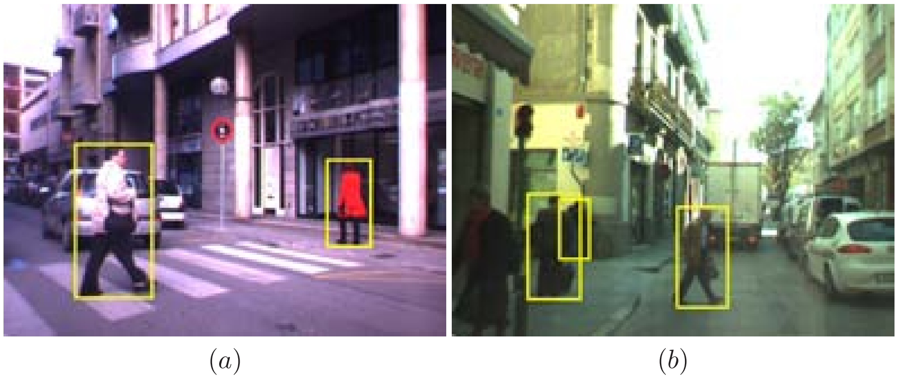


**Figure B.2:** Two images from CVC-02-CG with their corresponding annotations. As can be seen, partially occluded pedestrians are considered but not the ones with more than 15% of their annotation area out of the image, as in the left bottom corner of (*b*).

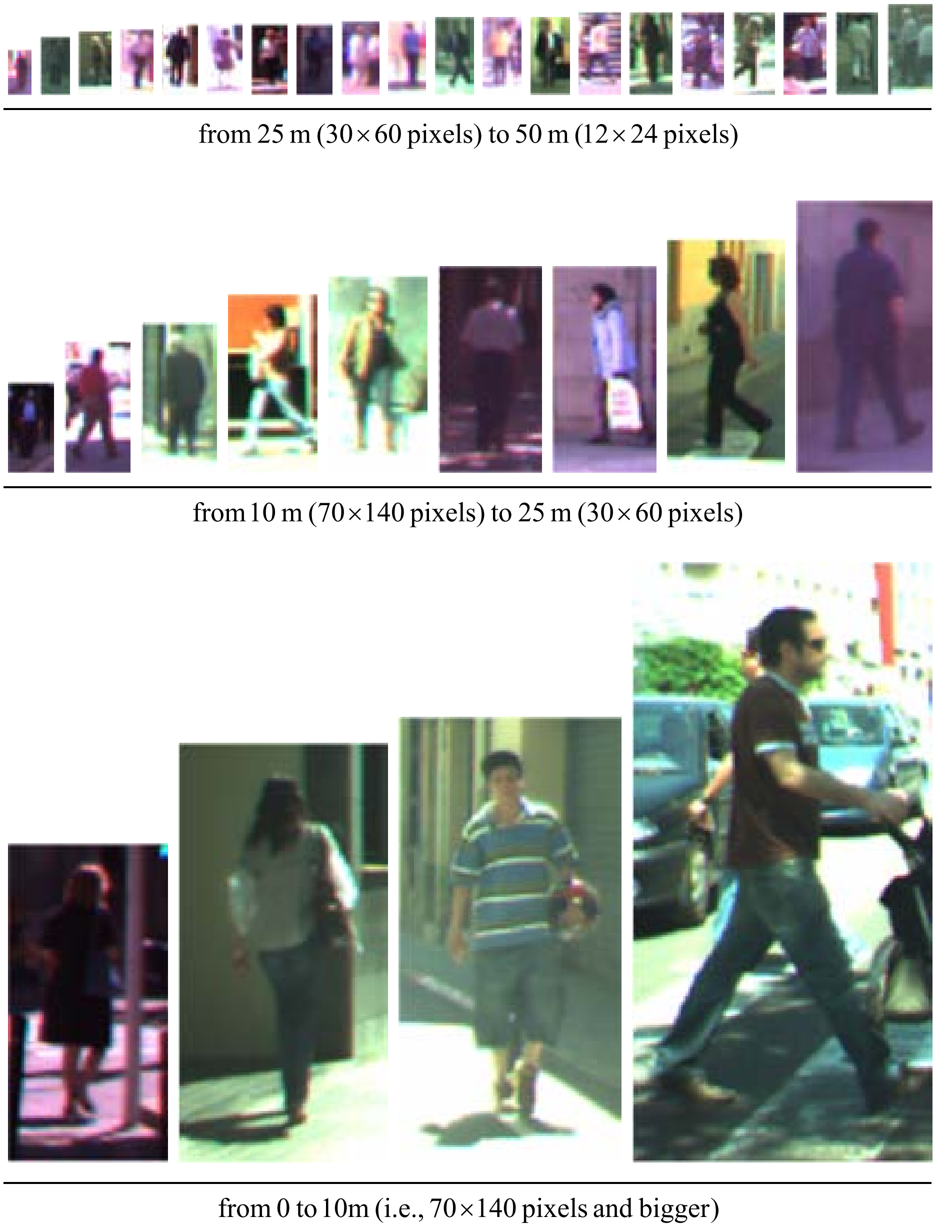
  - 153 pedestrian-free frames to select training negatives.

- The testing set for window-based evaluation consists of:
  - 570 cropped positives and their mirrors (original scale and 64 × 128).
  - 7500 cropped negatives using Sliding Window (original scale and 64 × 128).
  - 7500 cropped negatives using Adaptive Road Scanning (original scale and 64 × 128).
  - 150 pedestrian-free frames to select testing negatives.

- The testing set for image-based evaluation consists of:
  - 250 frames with 587 annotated pedestrians (57 are optional), from which 355 are near (0 to 25m) and 175 are far (25 to 50m).

## B.3 System (CVC-02-System)

The whole system evaluation in Chap. 6 consists of 15 sequences.



from 25 m (30×60 pixels) to 50 m (12×24 pixels)

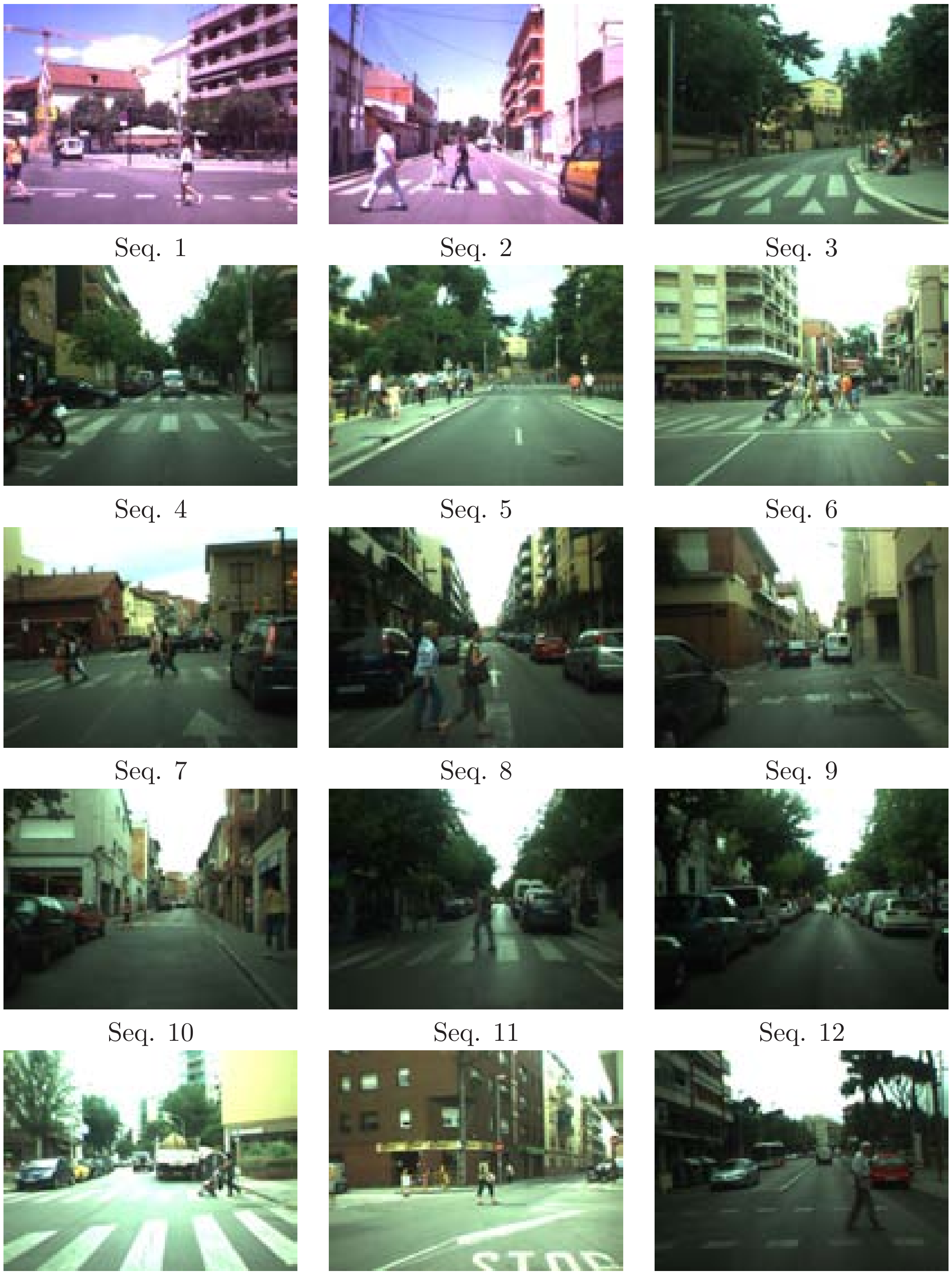
from 10 m (70×140 pixels) to 25 m (30×60 pixels)

from 0 to 10m (i.e., 70×140 pixels and bigger)

**Figure B.3:** Examples of CVC-02-Classification dataset.

**Table B.1:** CVC-02-System sequences. *Near* are pedestrians from 0 to 25 m, *Far* are pedestrians from 25 to 50m. Properties columns: column **P**edestrians (**I**solated or **C**rowded), column **B**ackground (**S**tructured, e.g., with parked vehicles, or **C**luttered) and **R**oad Slope (**C**onstant or **V**ariable).

| **Seq.** | **#Frames** | **#Annotations** | **#Near** | **#Far** | **Properties** | | |
|---|---|---|---|---|---|---|---|
| | | (Optional) | | | P | B | R |
| 1 | 546 | 888 (63) | 368 | 457 | I/C | C | C |
| 2 | 81 | 147 (5) | 128 | 14 | I | S | C |
| 3 | 304 | 1 114 (77) | 417 | 620 | C | C | V |
| 4 | 436 | 251 (7) | 73 | 171 | I | S | C |
| 5 | 185 | 862 (145) | 295 | 422 | I/C | C | V |
| 6 | 334 | 1 597 (136) | 812 | 649 | C | C | V |
| 7 | 566 | 1 044 (45) | 717 | 282 | I/C | S | V |
| 8 | 201 | 383 (4) | 210 | 169 | I | S | V |
| 9 | 244 | 262 (2) | 126 | 134 | I | S | V |
| 10 | 351 | 498 (14) | 329 | 155 | I | S | V |
| 11 | 151 | 151 (3) | 47 | 101 | I | S | C |
| 12 | 248 | 161 (47) | 83 | 31 | I | S | C |
| 13 | 115 | 82 (1) | 49 | 32 | I | C | C |
| 14 | 151 | 241 (12) | 147 | 82 | I | C | C |
| 15 | 451 | 302 (6) | 129 | 167 | I | S | V |
| Total | 4 364 | 7 983 (567) | 3 930 | 3 486 | | | |



Seq. 1

Seq. 2

Seq. 3

Seq. 4

Seq. 5

Seq. 6

Seq. 7

Seq. 8

Seq. 9

Seq. 10

Seq. 11

Seq. 12

Seq. 13

Seq. 14

Seq. 15

**Figure B.4:** Representative frames from each sequence of CVC-02-System.

# Appendix C

## State of the art tables

**Table C.1:** Visible Spectrum.

| Authors | Foreground Seg. | Object Classification | Verification/Refinement | Tracking | Sensor |
|---|---|---|---|---|---|
| *Gavrila et al.* (1999–2006) [66, 65, 67] | Stereo+PSC | Hierarchical Template Matching (Chamfer System [65])<br>NN-RBF based on texture [114] | Stereo | $\alpha - \beta$ tracker [67]<br>Silhouette, texture, stereo, fed to particle filters [69] | stereo |
| *Bertozzi et al.* (2003–04) [13, 16] | Symmetry+PSC | | Stereo, PSC, ad hoc image filters, 3D curves matching | Kalman, grey-level, stereo | stereo |
| *Broggi et al.* (2000–03) [26, 27] | Stereo (v-disparity), PSC, Symmetry, head and shoulders model matching. | | Stereo, PSC, entropy | — | stereo |
| *Shashua et al.* (2004) [141] | Texture +PSC | Parts-based gradient orientation and magnitude + Ridge Regression and AdaBoost | Multiframe after tracking: gait pattern, inward motion analysis scores (coupled with egomotion), goodness of classification | (used but not detailed) | mono |
| *Grubb et al.* (2004) [72] | Stereo (v-disparity) | Gradient magnitude+ Quadratic SVM | Classification goodness over time with help of tracking | Kalman+stereo | stereo |
| *Zhao et al.* (2000) [174] | Stereo+PSC | Gradient magnitude+ Neural Networks | — | — | stereo |
| *Soga et al.* (2005) [144] | Stereo+PSC | Four Directional Features (FDF) +Gaussian Kernel SVM | — | Extended Kalman Filter | stereo |
| *Gerónimo et al.* (2006-2007) [68] | Stereo-based Road Estimation [134] + PSC | Haar Wavelets and Edge Orientation Histograms + Real AdaBoost | — | — | stereo |

**Table C.2:** Visible Spectrum (continued).

| **Authors** | **Foreground Segmentation** | **Object Classification** | **Verification/Refinement** | **Tracking** | **Sensor** |
|---|---|---|---|---|---|
| *Parra et al.*<br>(2005-2007)<br>[125] | Stereo matching + lanes and road surface filtering + subtractive clustering | Parts-based SVM using edges and graylevel orientation, magnitude and texture | — | Kalman | stereo |
| *Leibe et al.*<br>(2005)<br>[90] | Scale-invariant Top-down Implicit Shape Model [91, 93] | | Combination of Chamfer matching and its overlap with the segmentation | — | mono |
| *Elzein et al.*<br>(2003)<br>[42] | Interframe motion and optical flow | Euclidean distance of Haar Wavelets between templates and ROI | — | — | mono |
| *Papageorgiou et al.*<br>(1997–2001)<br>[124, 112] | — | - Holistic: Haar W.+ Q.SVM [124]<br>- Parts-based: Haar W.+ Q/L.SVM [112] | — | Heuristic integration through time | mono |
| *Dalal/Zhu et al.*<br>(2005/06)<br>[35, 175] | — | Histograms of Oriented Grad. +<br>- Linear SVM [35]<br>- Linear SVM and AdaBoost cascade [175] | — | — | mono |
| *Wu et al.*<br>(2007)<br>[166] | — | Part-based multi-view edgels + Nested Weak Classifiers AdaBoost [77] Bayes-based parts combination | | Parts+color+ appearance-based Mean-shift | mono |

**Table C.3:** Infrared Spectrum.

| **Authors** | **Foreground Segmentation** | **Object Classification** | **Verification/Refinement** | **Tracking** | **Sensor** |
|---|---|---|---|---|---|
| *Broggi et al.* (2006) [29] | Double threshold and binarization | Geometrical moments (eccentricity, object and legs inclination) model matching using probability | — | — | mono (NIR) |
| *Sun et al.* (2004) [146] | Threshold | Polar coordinates shape + SVM based on gray-level image features + parts-based classif. | — | — | mono (NIR) |
| *Andreone et al.* (2005) [2] | PSC | Heuristic pixel-based discarding process + SVM using Haar wavelets | — | — | mono (NIR) |
| *Fang et al.* (2004) [49] | Vertical projection<br>Object contours | Histogram-, inertia-, contrast-based matching versus one unique pedestrian template | — | — | mono (TIR) |
| *Bertozzi et al.* (2003–05) [15, 12, 19] [20] | Hot areas, greyscale/edges vertical symmetry + Ad hoc filters + PSC [12, 15] | | 2D [12] / 3D [15, 28] model matching | Kalman filter with past history analysis [20] | mono (TIR) |
| | Horizontal/vertical histogram + Stereo [19] | | | | stereo |
| *Suard et al.* (2006) [145] | Simple threshold and bounding box generation | Histograms of Oriented Gradients + Linear SVM | — | — | mono (TIR) |
| *Mählisch et al.* (2005) [101] | PSC + Hypermutation Network (pixel classification [119]) | Fusion of Chamfer Contour Matching and a Haar W.-based classifier | Multiple filter approach based on area overlapping windows | Particle filter | mono (TIR) |
| *Xu et al.* (2005) [168] | Threshold on histogram-equalized image + PSC + ground extraction over edges map | Intensity image + SVM (head/entire body based) | — | Kalman, Mean-Shift Method | mono (TIR) |
| *Tsuji et al.* (2002) [154] | Threshold: intermediate value of brightness distribution histogram | — | — | Stereo + egomotion (yaw sensor) | stereo (TIR) |

**Table C.4:** Sensor fusion.

| Authors | Foreground Segmentation | Object Classification | Verification/Refinement | Tracking | Sensor |
|---|---|---|---|---|---|
| *Fardi et al.* (2005) [50] | Laserscanner map + Active contour on TIR to extract shape | Euclidean distance of Fourier descriptors between object and reference sets (on TIR) | Walking cycle analysis using optical flow and egomotion sensor | Kalman filter + data fusion | TIR (mono)<br>+ laser |
| *SAVE-U* (2005) [103] | Radar + Vision-based Local Histograms on Edge Images (visible and TIR) and PSC | NN-RBF [114] | Fusion of radar and tracking info | $\alpha$–$\beta$ tracker | TIR + visible (mono)<br>+ radar |
| *Milch et al.* (2001) [110] | Radar + velocity and steering sensors | Active contours (TIR or visible) using a trained shape model | — | — | TIR or visible (mono)<br>+ radar |
| *Bertozzi et al.* (2006-2007) [17, 14, 18] | v-disparity<br>+ disparity image<br>+ hotspots (TIR) | Symmetry + template matching + vertical edge-based refinement | | — | TIR + visible (both stereo) |
| *Premebida et al.* (2007) [130] | Laserscanner-based tracking of points | GMM on laserscanner data + AdaBoost (Haar Wav.) + Bayesian fusion of Classifiers | — | Kalman on laserscanner performed as foreground segmentation | Visible (mono)<br>+ laserscanner |

# Appendix D

# Performance evaluation

The statistical validation of any decision process is crucial to determine the performance of a wide variety of applications, for example from medical treatments to spam filtering or object detection. Let us define the null hypothesis as the default state of some phenomenon, for instance that *a patient does not have a disease* or that *a given window in an image contains just background clutter*. If we label the natural state of the null hypothesis as a negative, then the opposite state (i.e., *the patient does have a disease* and *a pedestrian has been found in the image*) is referred to as a positive. Since the decision process is aimed at rejecting or not rejecting the null hypothesis, then there exist two basic sources of error:

- false positives (FP) when the null hypothesis is incorrectly rejected (i.e., finding a disease in a healthy patient or detecting a pedestrian in a background image), or
- false negatives (FN) when the null hypothesis is incorrectly not rejected (i.e., finding healthy an ill patient or failing to find a pedestrian when in fact there is one).

Given these two errors, the performance measurement of a decision process consists in counting the number of FP and FN in the context of for example the total number of real positives or negatives, the total number of decisions, etc. On the contrary, a true positive (TP) is found if the null hypothesis is correctly rejected, whereas a true negative (TN) is found when the null hypothesis is correctly not rejected. Figure D.1 illustrates these concepts.

## D.1 Performance plots

Although basic measurements can be defined as real numbers, for example in terms of true positive rate (number of true positives out of the total number of positives), most classification algorithms typically provide a confidence value that shall be thresholded to take a decision. Thus, by varying such threshold we can plot different curves that show the classifiers performance in terms of the behavior of the basic measurements.

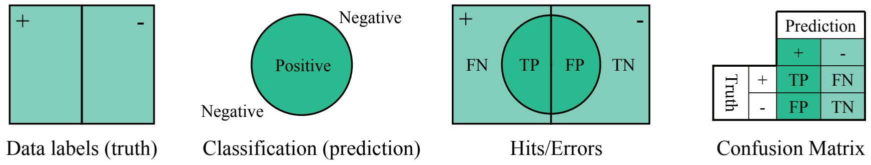


**Figure D.1:** Graphical representation and confusion matrix of errors and hits in the context of truth and prediction.

The **Receiver Operating Characteristic (ROC)** curve takes two measures into account, false positive rate (FPR) and true positive rate (TPR):

$$FPR = \frac{FP}{TN + FP} \quad , \tag{D.1}$$

$$TPR = \frac{TP}{TP + FN} \quad , \tag{D.2}$$

where FPR is plotted on the $x$-axis and TPR on the $y$-axis. The perfect classifier would have TPR=1 and FPR=0, placing the performance point at the top-left corner of the ROC. This curve has been used in a large number of papers [114, 68]. Dalal et al. [35] makes use of a complementary curve, called **Detection Error Trade-off (DET)**, which plots Miss Rate ($MR = 1 - TPR$) on the $y$-axis and FPR on $x$-axis, both axes using a logarithmic scale instead of a linear one.

Another widely used plot is **Recall-Precision (RP)** curve:

$$Recall = TPR \tag{D.3}$$

$$Precision = \frac{TP}{TP + FP} \quad , \tag{D.4}$$

with Recall on the $x$-axis and Precision on the $y$-axis, although sometimes their positions are interchanged and 1−Precision is used instead of the regular Precision. The perfect classifier in this case would have Recall= 1 and Precision= 1, which means that neither false positives nor false negatives exist. For instance, this curve is used in [47].

The final curve is **Sensitivity-Specificity**, defined as

$$Sensitivity = TPR \tag{D.5}$$

$$Specificity = \frac{TN}{TN + FP} \quad , \tag{D.6}$$

plotting sensitivity on $x$-axis and specificity on $y$-axis. Again, both Sensitivity and Specificity would be 1 in a perfect classifier.

All these measurements are summarized in Fig. D.2.

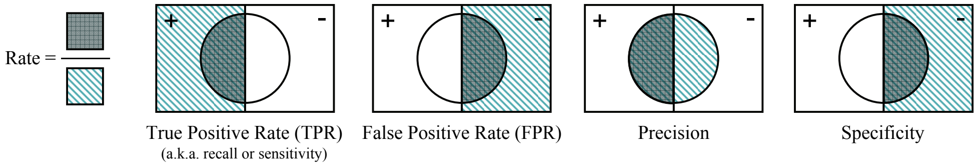


**Figure D.2:** Performance measures used in ROC, recall-precision and sensitivity-specificity curves.

## D.2 Window-based versus image-based evaluation

The common procedure while evaluating classifiers (either for medical tests or object classification, for instance) is to select a training set containing both positive and negative samples and a testing set with also positives and negatives. In order to plot a typical curve like ROC we just have to count the number of TP, FN, TN and FP on the testing set and compute the corresponding rates. This is possible because all the measures are well defined. In the case of pedestrian classification, a common procedure is to work with cropped images, say 1 000 test positives and 5 000 test negatives. However, some authors (e.g., [35]) just provide pedestrian-free images to randomly crop (both train and test) negatives, so the set is not well defined, that is, it is difficult to reproduce the exact results that the researchers present. In this case, different authors count False Positives Per Window (FPPW), which is equivalent to FPR but leave open the number of test negatives. If a range of confidence testing thresholds to count TPR and FPPW is used, this performance measure leads to a ROC in which the axes are TPR and FPPW, this latter one often using a logarithmic scale. We call this evaluation **window-based**. As it is made in the literature, we analyze the whole curve but special attention will be put to the point of the curve in which FPPW= $10^{-}4$, which represents one false positive each 10 000 tested negatives.

Another type of evaluation is focused on detection rather than classification, i.e., placing the performance in the context of frames rather than on isolated examples. In this case, since there are not cropped samples anymore, the way that a detection is considered as a TP or FP depends on the similarity of the detection with the annotations of a set of test frames. In this case, a ROC has the axes TPR and false positives per image (FPPI). The most used similarity measure at this moment is the detection-annotation overlap proposed by Everingham et al. [47]: a detection $d$ is marked as TP if its overlap with a window annotation $a$ exceeds a certain threshold $\Gamma$, where

$$\text{overlap}(d, a) = \frac{\text{area}(a \cap d)}{\text{area}(a \cup d)} \ , \tag{D.7}$$

otherwise the detection is marked FP. This evaluation is called **image-based**. In this case, the curve shall be read in a more global manner than in the image-based in the sense that the preferred FPPI working range will depend on the requirements for the given module/system. Throughout the thesis, and specially in Chap. 4, we focus on the DR value when FPPI= $10^0$, which corresponds to one average single false positive per frame, which is assumable for a system consisting of a classifier and cluster given that a tracking process is likely to absorb most of the spurious false positives.

In this thesis, the case in which multiple detections fulfilling this criterion for a single annotation is treated differently if we evaluate the classifier or the whole system (also the clustering). For example, in the case of the system, each annotation account for just a TP, the additional detections associated to the annotation are marked as FP, whereas in the case of the classification they are all marked as TP since there is not any clustering technique involved. This is detailed in the corresponding chapters.

# List of Acronyms

| | |
|---|---|
| ABS | antilock braking system |
| ACC | adaptive cruise control |
| ADAS | advanced driver assistance systems |
| AFL | advanced front lighting |
| CGP | candidate generation performance |
| CPA | candidates per annotation |
| DR | detection rate |
| EOH | edge orientation histograms |
| ESC | electronic stability control |
| FN | false negatives |
| FP | false positives |
| FPPI | false positives per image |
| FPPW | false positives per window |
| FPR | false positives rate |
| HF | Haar feature |
| HFOV | horizontal field of view |
| HOG | histograms of oriented gradients |
| ii | integral image |
| iv | integral volume |
| NIR | near infrared |
| NN | neural network |
| NPC | non pedestrian candidates |
| PPS | pedestrian protection system |
| ROI | region of interest |

| | |
|---|---|
| SIFT | scale invariant feature transform |
| SVM | support vector machine |
| SHOG | simplified HOG |
| | |
| TIR | thermal infrared |
| TN | true negatives |
| TP | true positives |
| TPR | true positives rate |
| | |
| VFOV | vertical field of view |
| VS | visible spectrum |

# Bibliography

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] L. Andreone, F. Bellotti, A. De Gloria, and R. Lauletta. SVM-based pedestrian recognition on near–infrared images. In *Proc. 4th Int. Symp. on Image and Signal Processing and Analysis*, pages 274–278, Zagreb, Croatia, 2005.

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[4] A.-S. Karlsson and A. Sjögren and L. Löwenadler and J. Hoetzel and E. Bianco and M. Miglieta and H.-J. Herzog and V. Willhoeft and M. Bertozzi, Deliverable D50.30. User needs state of the art and relevance for accidents. PReVENT project APALACI: Preventive and Active Safety Applications. 2005.

[5] R. Arndt, R. Schweiger, W. Ritter, D. Paulus, and O. Lhlein. Detection and tracking of multiple pedestrians in automotive applications. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 13–18, 2007.

[6] S.J. Ashton and G.M. Mackay. Benefits from changes in vehicle exterior design. *Proceedings of the Society of Automotive Engineers*, pages 255–264, 1983.

[7] Regional Development Australian Government Dept. of Infrastructure, Transport and Local Government. *Road Deaths in Australia – Monthly Bulletin.* 2009.

[8] Various Authors. *The World Health Report 2002 – Reducing risks, promoting healthy life.* World Health Organization, Geneva, Switzerland, 2002.

[9] AWAKE. `http://www.awake-eu.org`.

[10] H. Badino, W. Franke, and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Proc. Int. Conf. on Computer Vision, Workshop on Dynamical Vision*, Rio de Janeiro, Brazil, 2007.

[11] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[12] M. Bertozzi, A. Broggi, M. Carletti, A. Fascioli, T. Graf, P. Grisleri, and M.-M. Meinecke. IR pedestrian detection for advanced driver assistance systems. In *Proc. Pattern Recognition Symp.*, volume 2781, pages 582–590, Germany, 2003.

[13] M. Bertozzi, A. Broggi, R. Chapuis, F. Chausse, A. Fascioli, and A. Tibaldi. Shape–based pedestrian detection and localization. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 328–333, Shangai, China, 2003.

[14] M. Bertozzi, A. Broggi, M. Del Rose, and M. Felisa. A symmetry-based validator and refinement system for pedestrian detection in far infrared images. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 155–160, Seattle, WA, USA, 2007.

[15] M. Bertozzi, A. Broggi, A. Fascioli, T. Graf, and M.-M. Meinecke. Pedestrian detection for driver assistance using multiresolution infrared vision. *IEEE Trans. on Vehicular Technology*, 53(6):1666–1678, 2004.

[16] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse. Pedestrian localization and tracking system with Kalman filtering. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 584–589, Parma, Italy, 2004.

[17] M. Bertozzi, A. Broggi, M. Felisa, G. Vezzoni, and M. Del Rose. Low–level pedestrian detection by means of visible and far infra–red tetra–vision. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 231–236, Tokyo, Japan, 2006.

[18] M. Bertozzi, A. Broggi, C. Hilario, R.I. Fedriga, G. Vezzoni, and M . Del Rose. Pedestrian detection in far infrared images based on the use of probabilistic templates. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 327–332, Istanbul, Turkey, 2007.

[19] M. Bertozzi, A. Broggi, A. Lasagni, and M. Del Rose. Infrared stereo vision–based pedestrian detection. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 24–29, Las Vegas, NV, USA, 2005.

[20] E. Binelli, A. Broggi, A. Fascioli, S. Ghidoni, P. Grisleri, T. Graf, and M.-M. Meinecke. A modular tracking system for far infrared pedestrian recognition. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 759–764, Las Vegas, NV, USA, 2005.

[21] C. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995.

[22] C. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[23] R. Bishop. *Intelligent Vehicle Technologies and Trends.* Artech House, Inc., 2005.

[24] L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani. An evaluation of monocular image stabilization algorithms for automotive applications. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 1562–1567, Toronto, Canada, 2006.

[25] A. Broggi, M. Bertozzi, and A. Fascioli. Self–calibration of a stereo vision system for automotive applications. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3698–3703, Seoul, Korea, 2001.

[26] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. Shape–based pedestrian detection. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 215–220, Dearborn, MI, USA, 2000.

[27] A. Broggi, A. Fascioli, I. Fedriga, A. Tibaldi, and M. Del Rose. Stereo–based preprocessing for human shape localization in unstructured environments. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 410–415, Columbus, OH, USA, 2003.

[28] A. Broggi, A. Fascioli, P. Grisleri, T. Graf, and M.-M. Meinecke. Model–based validation approaches and matching techniques for automotive vision based pedestrian detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 3, page 1, San Diego, CA, USA, 2005.

[29] A. Broggi, R.I. Fedriga, A. Tagliati, T. Graf, and M.-M. Meinecke. Pedestrian detection on a moving vehicle: an investigation about near infra-red images. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 431–436, Tokyo, Japan, 2006.

[30] A. Broggi, P. Grisleri, T. Graf, and M.-M. Meinecke. A software video stabilization system for automotive oriented applications. In *Proc. Vehicular Technology Conf.*, volume 5, pages 2760–2764, Dallas, TX, USA, 2005.

[31] C.-Y. Chan and F. Bu. Literature review of pedestrian detection technologies and sensor survey. Technical report, Institute of Transportation Studies, Uni. of California at Berkeley, 2005.

[32] H. Caballero, A.M. López, and D. Gerónimo. Pedestrian detection refinement via non-explicit shape models. Technical report, Computer Vision Center, Universitat Autònoma de Barcelona, 2009.

[33] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(2):281–288, 2003.

[34] N. Dalal. *Finding People in Images and Videos.* PhD Thesis, Institut National Polytechnique de Grenoble / INRIA Rhône-Alpes, 2006.

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

[36] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. the European Conf. on Computer Vision*, volume 3952, pages 428–441, Graz, Austria, 2006.

[37] T. Dang and C. Hoffmann. Stereo calibration in vehicles. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 268–273, Parma, Italy, 2004.

[38] J. David and M. Keck. A two-stage approach to person detection in thermal imagery. In *Proc. Workshop on Applications of Computer Vision*, volume 1, pages 364–369, Breckenridge, CO, USA, 2005.

[39] E.D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of the SPIE Conference on Mobile Robots*, volume 727, pages 161–168, 1986.

[40] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu. Multiple component learning for object detection. In *Proc. the European Conf. on Computer Vision*, pages 211–224, Marseille, France, 2008.

[41] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: a benchmark. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 304–311, Miami Beach, FL, USA, 2009.

[42] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 500–504, Columbus, OH, USA, 2003.

[43] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence(in press)*, 31(12):2179–2195, 2008.

[44] M. Enzweiler and D.M. Gavrila. A mixed generative-discriminative framework for pedestrian classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[45] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[46] A. Ess, B. Leibe, and L. Van Gool. Depth and appearance for mobile scene analysis. In *Proc. Int. Conf. on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[47] M. Everingham, A. Zisserman, C. Williams, L. van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. García, T. Griffiths, F. Jurie, D. Keysers, M. Koskela v J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, , and J. Zhang. The 2005 pascal visual object classes challenge. In *Proceedings of the First PASCAL Challenges Workshop, LNAI, Springer-Verlag*, 2006.

[48] I. Fallon and D. O'neill. The world's first automobile fatality. *Accident Analysis and Prevention*, 37, 2005.

[49] Y. Fang, K. Yamada, Y. Ninomiya, B. Horn, and I. Masaki. A shape-independent method for pedestrian detection with far-infrared images. *IEEE Trans. on Vehicular Technology*, 53(6):1679–1697, 2004.

[50] B. Fardi, U. Schuenert, and G. Wanielik. Shape and motion-based pedestrian detection in infrared images: a multi sensor approach. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 18–23, Las Vegas, NV, USA, 2005.

[51] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[52] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381–395, June 1981.

[53] `http://www.fnir.eu`.

[54] Economic Comision for Europe. *Statistics of Road Traffic Accidents in Europe and North America*, volume LI. United Nations, 2007.

[55] National Highway Traffic Safety Administration National Center for Statistics and Analysis. *Traffic Safety Facts*. 2007.

[56] D. Forsyth, O. Arikan, L. Ikemoto, J. O' Brien, and D. Ramanan. *Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis*. Now publishers, 2005.

[57] U. Franke, D.M. Gavrila, S. Görzig, F. Lindner, F. Paetzold, and C. Wöhler. Autonomous driving goes downtown. *IEEE Intelligent Systems*, 13(6):40–48, 1999.

[58] U. Franke and S. Heinrich. Fast obstacle detection for urban traffic situations. *IEEE Trans. on Intelligent Transportation Systems*, 3(3):173–181, 2002.

[59] U. Franke and A. Joos. Real-time stereo vision for urban traffic scene understanding. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 273–278, Dearborn, MI, USA, 2000.

[60] U. Franke and I. Kutzbach. Fast stereo based object detection for Stop & Go traffic. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 339–344, Tokyo, Japan, 1996.

[61] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[62] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[63] S. Gammeter, A. Ess, T. Jäggli, K. Schindler, B. Leibe, and L. Van Gool. Articulated multi-body tracking under egomotion. In *Proc. the European Conf. on Computer Vision*, Marseille, France, 2008.

[64] T. Gandhi and M.M. Trivedi. Pedestrian protection systems: issues, survey, and challenges. *IEEE Trans. on Intelligent Transportation Systems*, 8(3):413–430, 2007.

[65] D.M. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. the European Conf. on Computer Vision*, volume 2, pages 37–49, Dublin, Ireland, 2000.

[66] D.M. Gavrila, J. Giebel, and S. Munder. Vision–based pedestrian detection: The PROTECTOR system. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 13–18, Parma, Italy, 2004.

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

[68] D. Gerónimo, A.D. Sappa, A.M. López, and D. Ponsa. Adaptive image sampling and windows classification for on-board pedestrian detection. In *Proc. 5th Int. Conf. on Computer Vision Systems*, Bielefeld, Germany, 2007.

[69] J. Giebel, D.M. Gavrila, and C. Schnör. A bayesian framework for multi–cue 3D object tracking. In *Proc. the European Conf. on Computer Vision*, pages 241–252, Prague, Czech Republic, 2004.

[70] E. Goubet, J. Katz, and F. Porikli. Pedestrian tracking using thermal infrared imaging. In *SPIE Conf. Infrared Technology and Applications*, volume 6206, pages 797–808, 2006.

[71] T. Graf, K. Seifert, M.-M. Meinecke, and R. Schmidt. Human factors in designing advanced night vision systems. In *5th Congress and Exhibition on Intelligent Transport Systems and Services*, Hannover, Germany, 2005.

[72] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe. 3D vision sensing for improved pedestrian safety. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 19–24, Parma, Italy, 2004.

[73] B. Hidalgo-Sotelo, A. Oliva, and A. Torralba. Human learning of contextual priors for object search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 86–93, Bellingham, WA, USA, 2005.

[74] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2137–2144, New York, NY, USA, 2006.

[75] D. Hoiem, A. Efros, and M. Hebert. Closing the loop in scene interpretation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[76] Z. Hu and K. Uchimura. U-V-disparity: an efficient algorithm for stereovision based scene analysis. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 48–54, Las Vegas, NV, USA, 2005.

[77] C. Huang, H. Ai, B. Wu, and S. Lao. Boosting nested cascade detector for multi-view face detection. In *Proc. Int. Conf. in Pattern Recognition*, volume 2, pages 415–418, 2004.

[78] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. the European Conf. on Computer Vision*, pages 343–356, Copenhagen, Denmark, 1996.

[79] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11), 1998.

[80] IVsource. Finally! adaptive cruise control arrives in the usa. `http://www.ivsource.net/archivep/2000/sep/a000929_USacc.html`, 2000.

[81] IVsource. Iteris' lane departure warning system now available on mercedes trucks in europe, 2000.

[82] M. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[83] K.C. Fuerstenberg. Pedestrian protection using laserscanners. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 115–120, Vienna, Austria, 2005.

[84] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. Int. Conf. on Computer Vision*, pages 166–173, 2005.

[85] P.M. Knoll. HDR vision for driver assistance. In B. Hoefflinger, editor, *High-Dynamic-Range (HDR) Vision*. Springer Berlin Heidelberg, 2007.

[86] S.J. Krotosky and M.M. Trivedi. Multimodal stereo image registration for pedestrian detection. In *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, pages 109–114, Seattle, WA, USA, 2007.

[87] S.J. Krotosky and M.M. Trivedi. On color-, infrared-, and multimodal-stereo approaches to pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 8(4):619–629, 2007.

[88] R. Labayrade and D. Aubert. A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 31–36, Columbus, OH, USA, 2003.

[89] R. Labayrade, D. Aubert, and J.P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through ”v–disparity” representation. In *Proc. IEEE Intelligent Vehicles Symp.*, volume 2, pages 17–21, Versailles, France, 2002.

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

[91] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, 2004.

[92] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *Int'l. Journal on Computer Vision*, 77(1-3):259–289, 2008.

[93] B. Leibe and B. Schiele. Scale–invariant object categorization using a scale–adaptive mean–shift search. In *DAGM04*, volume 3175, pages 145–153, Tuebingen, Germany, 2004.

[94] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 878–885, Washington, DC, USA, 2005.

[95] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 53–60, Washington, DC, USA, 2004.

[96] Z. Li, Y. Sun, F. Liu, and W. Shi. An effective and robust pedestrians detecting algorithm & symposia. In *IEEE Trans. on Intelligent Transportation Systems*, pages 545–549, Beijing, China, 2008.

[97] Z. Lin and L.S. Davis. A pose-invariant descriptor for human detection and segmentation. In *Proc. the European Conf. on Computer Vision*, volume 4, pages 423–436, Marseille, France, 2008.

[98] D.T. Linzmeier, M. Skutek, M. Mekhaiel, and K.C.J. Dietmayer. A pedestrian detection system based on thermopile and radar sensor data fusion. In *Int. Conf. on Information Fusion*, volume 2, 2005.

[99] D.G. Lowe. Distinctive image features from scale–invariant keypoints. *Int'l. Journal on Computer Vision*, 60(2):91–110, 2004.

[100] M.A. Sotelo, D. Fernández, J.E. Naranjo, C. González, R. García, T. de Pedro, and J. Reviejo. Vision-based adaptive cruise control for intelligent road vehicles. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 64–69, Sendai, Japan, 2004.

[101] M. Mählisch, M. Oberländer, O. Löhlein, D.M. Gavrila, and W. Ritter. A multiple detector approach to low–resolution FIR pedestrian recognition. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 325–330, Las Vegas, NV, USA, 2005.

[102] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[103] P. Marchal, M. Dehesa, D.M. Gavrila, M.-M. Meinecke, N. Skellern, and R. Viciguerra. SAVE–U. final report. Technical report, Information Society Technology Programme of the EU, 2005.

[104] S. Marsi, G. Impoco, A. Ukovich, S. Carrato, and G. Ramponi. Video enhancement and dynamic range control of HDR sequences for automotive applications. *EURASIP Journal on Advances in Signal Processing*, 2007.

[105] O. Mateo and K. Otsuka. Real-time visual tracker by stream processing. *Journal of Signal Processing Systems*, 2008.

[106] P. McCorduck. *Machines Who Think*. Natick, MA: A K Peters, Ltd, 2004.

[107] U. Meis, M. Oberländer, and W. Ritter. Reinforcing the reliability of pedestrian detection in far-infrared sensing. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 779–783, Parma, Italy, 2004.

[108] F. Miau, C. Papageorgiou, and L. Itti. Neuromorphic algorithms for computer vision and attention. In *International Symposium on Optical Science and Technology*, volume 4479, pages 12–23, Bellingham, WA, USA, 2001.

[109] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int'l. Journal on Computer Vision*, 65(1/2):43–72, 2005.

[110] S. Milch and M. Behrens. Pedestrian detection with radar and computer vision. In *Proc. Conf. on Progress in Automobile Lighting*, Darmstadt, Germany, 2001.

[111] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision–based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[112] A. Mohan, C. Papageorgiou, and T. Poggio. Example–based object detection in images by components. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[113] D. Mohan. Traffic safety and health in indian cities. *Journal of Transport Infrastructure*, 9:79–94, 2002.

[114] S. Munder and D.M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, 2006.

[115] H. Nanda and L. Davis. Probabilistic template based pedestrian detection in infrared videos. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 15–20, Versailles, France, 2002.

[116] S.K. Nayar and V. Branzoi. Adaptive dynamic range imaging: optical control of pixel exposures over space and time. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1168–1175, Nice, France, 2003.

[117] S. Nedevschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt. High accuracy stereovision approach for obstacle detection on non–planar roads. *Proc. IEEE Intelligent Engineering Systems*, pages 211–216, 2004.

[118] J. Needham. *Science and Civilisation in China. Vol. 2, History of Scientific Thought.* Cambridge University Press, 1954.

[119] M. Oberländer. Hypermutation networks – a discrete approach to machine perception. In *Third Weightless Neural Networks Workshop*, University of York, United Kingdom, 2005.

[120] Government of India Department of road transport and highways. Total number of motor vehicles in india 1951-2004. `http://morth.nic.in/writereaddata/sublinkimages/table-12458822488.htm`, 2004.

[121] Organisation internationale des constructeurs d'automobiles. `http://www.oica.net`.

[122] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson. A new pedestrian dataset for supervised learning. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 373–378, Eindhoven, The Netherlands, 2008.

[123] J. Pang, Q. Huang, and S. Jiang. Multiple instance boost using graph embedding based decision stump for pedestrian detection. In *Proc. the European Conf. on Computer Vision*, volume 4, pages 541–552, Marseille, France, 2008.

[124] C. Papageorgiou and T. Poggio. A trainable system for object detection. *Int'l. Journal on Computer Vision*, 38(1):15–33, 2000.

[125] I. Parra, D. Fernández, M.A. Sotelo, L.M. Bergasa, P. Revenga, J. Nuevo, M. Ocaña, and M.A. García. Combination of feature extraction method for SVM pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 8(2):292–307, 2007.

[126] PASCAL Visual Object Classes Challenge. `http://pascallin.ecs.soton.ac.uk/challenges/VOC`.

[127] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A.A. Hyder, E. Jarawan, and C. Mathers. *World Report on road traffic injury prevention.* World Health Organization, Geneva, Switzerland, 2004.

[128] V. Philomin, R. Duraiswami, and L.S. Davis. Pedestrian tracking from a moving vehicle. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 350–355, Dearborn, MI, USA, 2000.

[129] D. Ponsa, A.M. López, J. Serrat, F. Lumbreras, and T. Graf. 3D vehicle sensor based on monocular vision. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 1096–1101, Vienna, Austria, 2005.

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

[131] PReVENT. `http://www.prevent-ip.org`.

[132] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection.* John Wiley & Sons, New York, 1987.

[133] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, 2007.

[134] A.D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A.M. López. An efficient approach to onboard stereo vision system pose estimation. *IEEE Trans. on Intelligent Transportation Systems*, 9(3):476–490, 2008.

[135] A.D. Sappa, R. Herrero, F. Dornaika, D. Gerónimo, and A. López. Road approximation in euclidean and v-disparity space: a comparative study. In *Computer Aided Systems Theory*, volume 4739, pages 1105–1112, 2007.

[136] `http://www.save-u.org/`.

[137] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[138] E. Seeman, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1582–1588, New York, NY, USA, 2006.

[139] E. Seeman and B. Schiele. Cross-articulation learning of robust detection of pedestrians. In *DAGM Symposium*, Berlin, Germany, 2006.

[140] N. Sharkey. The programmable robot of ancient greece. *New Scientist*, (2611):32–35, 2007.

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

[142] H. Shimizu and T. Poggio. Direction estimation of pedestrian from images. Technical report, Artificial Intelligence Group, Massachusetts Institute of Technology, 2003.

[143] V.K. Singh, B. Wu, and R. Nevatia. Pedestrian tracking by associating tracklets using detection residuals. In *Workshop on Motion and Video Computing*, pages 1–8, Copper Mountain, CO, USA, 2008.

[144] M. Soga, T. Kato, M. Ohta, and Y. Ninomiya. Pedestrian detection with stereo vision. In *Proc. IEEE Int. Conf. on Data Engineering Workshop*, page 1200, Tokyo, Japan, 2005.

[145] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 206–212, Tokyo, Japan, 2006.

[146] H. Sun, C. Hua, and Y. Luo. A multi–stage classifier based algorithm of pedestrian detection in night with a near infrared camera in a moving car. In *Proc. Third Int. Conf. on Image and Graphics*, pages 120–123, Hong Kong, China, 2004.

[147] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.

[148] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional neural networks. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 224–229, Las Vegas, NV, USA, 2005.

[149] Q. Tian, H. Sun, Y. Luo, and D. Hu. Nighttime pedestrian detection with a normal camera using SVM classifier. In *Int. symp. on neural networks*, pages 189–194, 2005.

[150] A. Torralba and A. Oliva. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007.

[151] Japanese National Police Agency Traffic Bureau. *Statistics Road Accidents Japan.* International Association of Traffic and Safety Sciences, 2007.

[152] D. Tran and D. Forsyth. Configuration estimates improve pedestrian finding. In *Conf. on Neural Information Processing Systems Conference*, pages 1529–1536, Vancouver, Canada, 2007.

[153] T.J. Triggs and W.G. Harris. Reaction time of drivers to road stimuli. Technical report, Monash University, Victoria, Australia, 1982.

[154] T. Tsuji, H. Hattori, M. Watanabe, and N. Nagaoka. Development of night-vision system. *IEEE Trans. on Intelligent Transportation Systems*, 3(3):203–209, 2002.

[155] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian Manifold. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10), 2008.

[156] W. van der Mark and D.M. Gavrila. Real–time dense stereo for intelligent vehicles. *IEEE Trans. on Intelligent Transportation Systems*, 7(1):38–50, 2006.

[157] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[158] P. Viola and M. Jones and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 734–741, 2003.

[159] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, HI, USA, 2001.

[160] L. Vlacic, M. Parent, and F. Harashima. *Intelligent Vehicle Technologies.* Butterworth-Heinemann, 2001.

[161] C. Wang, H. Tanahashi, H. Hirayu, Y. Niwa, and K. Yamamoto. Comparison of local plane fitting methods for range data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 663–669, Kauai Marriot, HW, USA, December 2001.

[162] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Department of Computer Science, 2002.

[163] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization: a parallel technique. In *Symposium of the German Association for Pattern Recognition*, pages 71–81, Munich, Germany, 2008.

[164] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In *DAGM Symposium*, pages 82–91, Munich, Germany, 2008.

[165] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *Proc. Int. Conf. on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[166] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. *Int'l. Journal on Computer Vision*, 75(2):247–266, 2007.

[167] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, 2007.

[168] F. Xu, X. Liu, and K. Fujimura. Pedestrian detection and tracking with night vision. *IEEE Trans. on Intelligent Transportation Systems*, 6(1):63–71, 2005.

[169] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *Proc. of the Pacific Asia Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, 2004.

[170] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[171] L. Zhang and R. Nevatia. Efficient scan-window based object detection using GPGPU. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.

[172] L. Zhang, B. Wu, and R. Nevatia. Pedestrian detection in infrared images based on local shape features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, 2007.

[173] W. Zhang, G. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *Proc. Int. Conf. on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, 2007.

[174] L. Zhao and C. Thorpe. Stereo and neural network–based pedestrian detection. *IEEE Trans. on Intelligent Transportation Systems*, 1(3):148–154, 2000.

[175] Q. Zhu, S. Avidan, M-C. Yeh, and K-T. Cheng. Fast human detection using a cascade of histrograms of oriented gradients. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1491–1498, New York City, NY, USA, 2006.

# Publications

## Journals

D. Gerónimo, A.M. López, A.D. Sappa and T. Graf, Survey of Pedestrian Detection for Advanced Driver Assistance Systems. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*in press*), 2009. (Impact Factor: 5.960, 1st/94 in Computer Science, Artificial Intelligence)

A.D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo and A.M. López. An Efficient Approach to Onboard Stereo Vision System Pose Estimation. In *IEEE Transactions on Intelligent Transportation Systems*, Vol.9, Num. 3, pp. 476-490. September 2008. (Impact Factor: 2.844, 1st/23 in Transportation Science & Technology)

A.D. Sappa, D. Gerónimo, F. Dornaika and A.M. López. On-board camera extrinsic parameter estimation. In *IEE Electronics Letters*, 42(13), pp. 745747. 2006. (Impact Factor: 1.140, 108th/226 in Engineering, Electrical & Electronic)

D. Gerónimo, A.D. Sappa, D. Ponsa and A.M. López. 2D-3D Based On-Board Pedestrian Detection System. *Computer Vision and Image Understanding* (Submitted on December 2007, waiting to Final Acceptance Decision after a Second Review).

## Book Chapters

D. Gerónimo, A.D. Sappa, A.M. López. Stereo-based candidate generation for pedestrian protection systems. In *Binocular Vision: Development, Depth and Disorders*, NOVA Publishers (*in press*), 2009.

A.D. Sappa, D. Gerónimo, F. Dornaika, A.M. López. Stereo Vision Camera Pose Estimation for On-Board Applications. Chapter 3, pp. 39-50., in *Scene Reconstruction, Pose Estimation and Tracking*, Ed. Rustam Stolking, 2007, ISBN-978-3-902613-06-6.

## International Conferences

A.D. Sappa, F. Dornaika, D. Gerónimo and A.M. López. Registration-based moving object detection from moving camera. In *International Conference on Intelligent Robots and Systems, 2nd Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pp.65-69. Nice, France, 2008.

D. Gerónimo, A.M. López and A.D. Sappa. Computer Vision Approaches for Pedestrian Detection: Visible Spectrum Survey. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, Part 1, LNCS 4477, pp. 547-554. Girona, Spain, June 2007.

D. Gerónimo, A.M. López, D. Ponsa and A.D. Sappa. Haar Wavelets and Edge Orientation Histograms for On-Board Pedestrian Detection. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, Part 1, LNCS 4477, pp. 418-425. Girona, Spain, June 2007.

D. Gerónimo, A.D. Sappa, A.M. López and D. Ponsa. Adaptive Image Sampling and Windows Classification for On-Board Pedestrian Detection. In *Proceedings of the International Conference on Computer Vision Systems*. Bielefeld, Germany, March, 2007.

A.D. Sappa, F. Dornaika, D. Gerónimo and A.M. López . Efficient On-Board Stereo Vision Pose Estimation. In *EUROCAST2007, Workshop on Cybercars and Intelligent Vehicles*, LNCS 4739, pp. 1183-1190. Las Palmas de Gran Canaria, Spain, February 2007.

A.D. Sappa, R. Herrero, F. Dornaika, D. Gerónimo and A.M. López . Road approximation in euclidean and v-disparity space: A comparative study. In *EUROCAST2007, Workshop on Cybercars and Intelligent Vehicles*, LNCS 4739, pp. 1105-1112. Las Palmas de Gran Canaria, Spain, February 2007.

A.D. Sappa, D. Gerónimo, F. Dornaika and A.M. López. Real time vehicle pose using on-board stereo-vision System. In *International Conference on Image Analysis and Recognition*, LNCS 4142, pp. 205-216. Pvoa de Varzim, Portugal, September 18-20, 2006.

D. Gerónimo, A.D. Sappa, A.M. López and D. Ponsa. Pedestrian detection using AdaBoost learning of features and vehicle pitch estimation. In *Proceedings of the IASTED International Conference on Visualization*, Imaging and Image Processing, pp. 400-405. Palma de Mallorca, Spain, August 28-30, 2006.