HAL
open science

Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments

R. Omar Chavez-Garcia

To cite this version:

R. Omar Chavez-Garcia. Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments. Robotics [cs.RO]. Université de Grenoble, 2014. English. NNT : . tel-01082021

HAL Id: tel-01082021

https://hal.science/tel-01082021v1

Submitted on 12 Nov 2014

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématique, Informatique (Robotique)**

Arrêté ministériel :

Présentée par

Ricardo Omar CHAVEZ GARCIA

Thèse dirigée par **Olivier AYCARD**

préparée au sein **Laboratoire d'Informatique de Grenoble**
et de **Mathématiques, Sciences et Technologies de l'Information, Informatique**

Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments

Thèse soutenue publiquement le **25 septembre 2014**,
devant le jury composé de :

M., Michel DEVY
LAAS-CNRS, Rapporteur
M., François CHARPILLET
INRIA Nancy, Rapporteur
Mme., Michelle ROMBAUT
Gipsa-Lab, Président, Examinatrice
M., Yassine RUICHEK
Université de Technologie de Belfort-Montbéliard, Examinateur
M., Olivier AYCARD
Université de Grenoble1, Directeur de thèse

Multiple Sensor Fusion for Detection, Classification and Tracking of Moving Objects in Driving Environments

To:

my family
and my friends.

Abstract

Advanced driver assistance systems (ADAS) help drivers to perform complex driving tasks and to avoid or mitigate dangerous situations. The vehicle senses the external world using sensors and then builds and updates an internal model of the environment configuration. Vehicle perception consists of establishing the spatial and temporal relationships between the vehicle and the static and moving obstacles in the environment. Vehicle perception is composed of two main tasks: simultaneous localization and mapping (SLAM) deals with modelling static parts; and detection and tracking moving objects (DATMO) is responsible for modelling moving parts of the environment. The perception output is used to reason and decide which driving actions are the best for specific driving situations. In order to perform a good reasoning and control, the system has to correctly model the surrounding environment. The accurate detection and classification of moving objects is a critical aspect of a moving object tracking system. Therefore, many sensors are part of a common intelligent vehicle system.

Multiple sensor fusion has been a topic of research since long; the reason is the need to combine information from different views of the environment to obtain a more accurate model. This is achieved by combining redundant and complementary measurements of the environment. Fusion can be performed at different levels inside the perception task.

Classification of moving objects is needed to determine the possible behaviour of the objects surrounding the vehicle, and it is usually performed at tracking level. Knowledge about the class of moving objects at detection level can help to improve their tracking, reason about their behaviour, and decide what to do according to their nature. Most of the current perception solutions consider classification information only as aggregate information for the final perception output. Also, the management of incomplete information is an important issue in these perception systems. Incomplete information can be originated from sensor-related reasons, such as calibration issues and hardware malfunctions; or from scene perturbations, like occlusions, weather issues and object shifting. It is important to manage these situations by taking into account

the degree of imprecision and uncertainty into the perception process.

The main contributions in this dissertation focus on the DATMO stage of the perception problem. Precisely, we believe that including the object's class as a key element of the object's representation and managing the uncertainty from multiple sensors detections, we can improve the results of the perception task, i.e., a more reliable list of moving objects of interest represented by their dynamic state and appearance information. Therefore, we address the problems of sensor data association, and sensor fusion for object detection, classification, and tracking at different levels within the DATMO stage. We believe that a richer list of tracked objects can improve future stages of an ADAS and enhance its final results.

Although we focus on a set of three main sensors: radar, lidar, and camera, we propose a modifiable architecture to include other type or number of sensors. First, we define a composite object representation to include class information as a part of the object state from early stages to the final output of the perception task. Second, we propose, implement, and compare two different perception architectures to solve the DATMO problem according to the level where object association, fusion, and classification information is included and performed. Our data fusion approaches are based on the evidential framework, which is used to manage and include the uncertainty from sensor detections and object classifications. Third, we propose an evidential data association approach to establish a relationship between two sources of evidence from object detections. We apply this approach at tracking level to fuse information from two track representations, and at detection level to find the relations between observations and to fuse their representations. We observe how the class information improves the final result of the DATMO component. Fourth, we integrate the proposed fusion approaches as a part of a real-time vehicle application. This integration has been performed in a real vehicle demonstrator from the *interactIVe* European project.

Finally, we analysed and experimentally evaluated the performance of the proposed methods. We compared our evidential fusion approaches against each other and against a state-of-the-art method using real data from different driving scenarios and focusing on the detection, classification and tracking of different moving objects: pedestrian, bike, car and truck. We obtained promising results from our proposed approaches and empirically showed how our composite representation can improve the final result when included at different stages of the perception task.

Key Words: Multiple sensor fusion, intelligent vehicles, perception, DATMO, classification, multiple object detection & tracking, Dempster-Shafer theory

Acknowledgements

I am highly thankful to my director of thesis and my colleges for all the advices and help.

This work was also supported by the European Commission under interactIVe, a large scale integrating project part of the FP7-ICT for Safety and Energy Efficiency in Mobility. I would like to thank all partners within interactIVe for their cooperation and valuable contribution.

Contents

1 Introduction **1**

1.1 Perception for intelligent Vehicles 4

1.2 Preprocessing 6

1.2.1 Sensor uncertainty 7

1.3 Simultaneous localization and mapping 8

1.4 Detection, Classification and Tracking of Moving Objects 10

1.4.1 Detection of moving objects 10

1.4.2 Tracking of moving objects 11

1.4.3 Classification 12

1.5 Simultaneous localization, mapping and moving object tracking 13

1.6 Multi-sensor fusion for vehicle perception 13

1.6.1 Requirements of multi-sensor systems 16

1.7 Contributions 16

1.8 Thesis outline 18

2 Intelligent Vehicle Perception **20**

2.1 Vehicle perception problem 20

2.2 Simultaneous Localization and Mapping 22

2.2.1 Map representation 24

2.3 Detection and Tracking of Moving Objects 25

2.3.1 Moving Object Detection 26

2.3.2 Tracking of Moving Objects 27

2.3.3 Data Association 29

2.3.4 Filtering 30
2.4 Object classification 32
2.4.1 Vehicle classification 34
2.4.2 Pedestrian classification 38
2.5 Multi-sensor fusion 42
2.5.1 Fusion methods 42
2.5.2 Fusion Architectures 53
2.6 Summary 57

3 Methodology overview **59**
3.1 interactIVe project 60
3.2 Perception subsystem 63
3.2.1 Perception architecture inside the *interactIVe* project 64
3.3 Vehicle demonstrator and sensor configuration 64
3.4 Fusion architectures 68
3.5 Sensor data processing 70
3.5.1 Lidar processing 70
3.5.2 Radar targets 77
3.5.3 Camera images 78
3.6 Summary 82

4 Multi-sensor fusion at tracking level **85**
4.1 Moving object detection, tracking and classification 87
4.1.1 Lidar sensor 88
4.1.2 Radar sensor 93
4.1.3 Camera Sensor 96
4.2 Multi-sensor fusion at tracking level 99
4.2.1 Instantaneous Combination 100
4.2.2 Dynamic Combination 103
4.3 Experimental Results 104

4.4 Summary 107

5 Multi-sensor fusion at detection level **110**

5.1 Moving object detection and classification 112

5.1.1 Lidar sensor 113

5.1.2 Radar sensor 114

5.1.3 Camera sensor 115

5.2 Fusion at detection level 116

5.2.1 Data association 117

5.2.2 Moving object tracking 121

5.3 Experimental results 122

5.4 Summary 127

6 Application: Accident Avoidance by Active Intervention for Intelligent Vehicles (interactIVe) **129**

6.1 Fusion approach integration 130

6.1.1 Modules architecture 130

6.2 Experimental evaluation 139

6.2.1 Real time assessment 140

6.2.2 Qualitative evaluation 140

6.2.3 Quantitative evaluation 142

6.3 Summary 144

7 Conclusion and Perspectives **147**

7.1 Conclusions 148

7.2 Perspectives 150

7.2.1 Short-term perspectives 150

7.2.2 Long-term perspectives 151

References **152**

List of Figures

1.1 Architecture for an autonomous robotic system. 5

1.2 Example output of a perception system into a car. Left: image capture of a real urban scenario. Center: Raw measurements from a lidar sensor. Right: Environment representation of the real scenario, static and moving objects are highlighted. 5

1.3 General architecture of the perception task and its two main components: simultaneous localization and mapping (SLAM) and detection and tracking of moving objects (DATMO). 6

1.4 Examples of the three main approaches to perform map representation. The representation is based on lidar measurements. 9

1.5 Architecture of the proposed approaches for data association and fusion at two different levels: (a) detection level, (b) tracking level. 17

2.1 Process diagram for intelligent vehicle perception. 21

2.2 Common architecture of a Moving Object Tracking module. 28

2.3 Graphical representation of the problem of multiple motion model object tracking (Vu, 2009). 32

2.4 Results of moving object tracking based on (left) a single motion model and (right) multiple motion models (Wang *et al.*, 2007). 32

2.5 Fusion levels within the SLAM and DATMO components interaction. . . 54

3.1 Objectives per vehicle demonstrators involved in the *interactIVe* project. 61

3.2 Layers inside the system architecture. 62

3.3 *interactIVe* generic architecture. 63

3.4 Schematic of the perception platform. 65

3.5 CRF demonstrator vehicle on Lancia Delta, with location of sensors and driver-interaction channels. Active safety belt is not shown. 67

3.6 Architecture of the system modules for the CRF demonstrator vehicle. . 67

3.7 Left: images of the CRF vehicle demonstrator. Right: Field of view of the three frontal sensors used as inputs to gather datasets for our proposed fusion approaches detailed in Chapter 4 and Chapter 5, and for the implementation of a real perception application presented in Chapter 6. . . 68

3.8 General overview of our two proposed fusion architectures at tracking (a) and at detection level (b). 70

3.9 *Sensor model* of a laser scanner, the curve models the probability of getting at distance d from the lidar position, z_{max} represents the maximum range of the scanner (left). *Inverse sensor model*, this curve models the probability of the presence of an object at different distances for which laser gives the measure z_t. (right) 71

3.10 Vehicle motion model $P(x_t|u_t, \hat{x}_{t-1})$ (left). Sampling of vehicle motion model (right). Vehicle position is considered to be at the center of the vehicle. Blur gray extremes represent the uncertainty in the two components of control input $u_t = (v_t, \omega_t)$. 75

3.11 Occupancy grid representation obtained by processing raw lidar data. From left to right: Reference image from camera; static occupancy grid M_{t-1} obtained by applying the SLAM solution; current lidar scan represented in an occupancy grid; detection of the moving objects (green bounding boxes). 77

3.12 Informative blocks for each object class detection window, from left to right: pedestrian, car, and truck (for sake of clarity, only some of them are displayed). Histograms of gradients are computed over these sparse blocks and concatenated to form S-HOG descriptors. Average size of the descriptors for pedestrians, cars and trucks are 216, 288 and 288 respectively. 80

3.13 Examples of successful detection of pedestrians (top) and cars (bottom) from camera images. 83

4.1 General architecture of the multi-sensor fusion approach at tracking level. 86

4.2 Architecture of the multi-sensor fusion approach at tracking level for lidar, camera and radar sensors. 86

4.3 Fitting model process: fixed object box model (green); L-shape and I-shape segments (red). 90

4.4 Output example from pedestrian (top) and vehicle classifiers (down) after being applied over the regions of interest from lidar moving object detection described in Section 4.1.1. 98

4.5 Schematic of the proposed fusion architecture. 99

4.6 Results from the proposed fusion approach for moving object classification in urban scenarios (a,b) . The left side of each image represents the top view representation of the scene (static and moving objects) showed in the right-side image. Bounding shapes and class tags in the right side of each image represent classified moving objects. 108

4.7 Results from the proposed fusion approach for moving object classification in (a,b) highway scenarios. The left side of each image represents the top view representation of the scene (static and moving objects) showed in the right-side image. Bounding shapes and class tags in the right side of each image represent classified moving objects. 109

5.1 General architecture of the fusion approach at detection level. 111

5.2 General architecture of the fusion approach at detection level for three main sensors: lidar, radar, and camera. 111

5.3 Results of the complete DATMO solution for urban areas. Several radar targets are discarded due to lack of support from the other sensor detections. The left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. The right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object. 123

5.4 Results of the complete DATMO solution for a highway scenario. Vehicles at high speeds are detected. The left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. The right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object. 124

5.5 Results of the complete DATMO solution for urban areas. Several objects of interest are detected. Left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. Right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object. 125

6.1 Integrated FOP-MOC module architecture. 131

6.2 FOP-MOC module output: detection, classification and tracking of cars on test track scenario. The test tracks scenario is part of the CRF facilities. 141

6.3 FOP-MOC module output: detection, classification and tracking of pedestrians on test track scenario. The test tracks scenario is part of the CRF facilities. 141

6.4 FOP-MOC module output: detection, classification and tracking of objects of interest in real scenarios. First case: highway. Second case: urban areas. 142

List of Tables

2.1 Feature comparison among the main map representations schemes. . . . 25

3.1 Implemented use cases on the CRF vehicle demonstrator for Continuous Support functions. 66

4.1 Number of car/truck miss-classifications from individual lidar and radar tracking modules and from the proposed fusion approach. 105

4.2 Number of pedestrian/bikes miss-classifications from individual lidar and radar tracking modules and from the proposed fusion approach.. Highway datasets do not contain any pedestrian or bike. 106

5.1 Number of vehicle (car and truck) mis-classifications obtained by the fusion approaches. 126

5.2 Number of pedestrian and bike mis-classifications obtained by the fusion approaches. Highway datasets do not contain any pedestrian or bike. 126

5.3 Number moving objects false detections obtained by the fusion approaches. 126

6.1 Quantitative results of the FOP-MOC module for four different scenarios: highway, urban area, rural road and test track. Four objects of interest are taken into account: (P) pedestrian, (B) bike, (C) car, and (T) truck. 145

CHAPTER 1

Introduction

ROBOTICS is the science of perceiving, modelling, understanding and manipulating the physical world through computer-controlled devices (Thrun, 2000). Some examples of robotic systems in use include mobile platforms for planetary exploration (Grotzinger *et al.*, 2012), robotic arms used for industrial purposes (Mason and Salisbury, 1985), intelligent vehicles powered by robotics while engaging in real scenarios (Montemerlo *et al.*, 2006), robotic systems for assisted surgeries (Gerhardus, 2003) and companion robots assisting humans with many common tasks (Coradeschi and Saffiotti, 2006). Although different, the robotic systems mentioned above have common applications in the physical world, the ability to perceive their surroundings through sensors, and to manipulate their environment through physical mechanisms.

Intelligent vehicles have moved from being a robotic application of tomorrow to a current area of extensive research and development. Their main objective is to provide safety whilst driving. The most striking characteristic of an intelligent vehicle system (from now on intelligent system), i.e., a system embedded into cars or trucks, is that it has to operate in increasingly unstructured environments, environments that are inherently uncertain and dynamic.

Safety while driving has been an important issue since the rise of the first automotive applications for driving assistance. Systems embedded in intelligent vehicles aim at providing a safe driving experience. These systems are built with the intervention of robotics, automotive engineering and sensor design. We can divide them according to the degree of immersion they have. They can focus on environment monitoring, to assist the driver with the vehicle navigation and warn him in dangerous scenarios. They can physically intervene to mitigate and avoid collisions or unsafe situations. Ultimately, they can be autonomous and take full control of the vehicle to drive as the user desires. Some of the tasks provided by an intelligent vehicle system

include: automated driving, navigation assistance, moving obstacles detection, traffic signal detection, warning notifications, breaking assistance and automated parking.

The ultimate intelligent vehicle system is one embedded into an autonomous vehicle, also known as a driver-less car. We expect to see it soon driving on common city, country or high-speed roads. Some of the tasks we want this vehicle to perform are: transport of people or goods from one point to another without human intervention, sharing the road with other autonomous or human-driven vehicles while following the traffic rules and protecting the safety of the people even in unexpected conditions. Autonomous vehicles applications have long been under research since several years ago, in the following we will mention some early milestones achieved in this area.

The *Stanford Cart* is considered as the first *intelligent* vehicle, built with some degree of autonomy in 1961, its goal was to control a moon rover from Earth using only a video camera (Adams, 1961). At that time the vehicle had no on-board computer and was connected to a control console through a long cable and carried a TV camera. By 1971, the vehicle was able to follow a high contrast white line under controlled lighting conditions using video image processing (Schmidt and Rodney, 1971). Another improvement was made around 1979, when the vehicle used binocular vision to navigate avoiding obstacles in a controlled indoor environment (Moravec, 1980). During the same years, in 1977 Japan's Tsukuba Mechanical Engineering Lab developed a vehicle capable of driving on a road at a speed of *30 km/h* by tracking street markers using only images from a video camera. In 1972, the Artificial Intelligence Center at SRI developed an indoor robotic system equipped with multiple sensors: a TV camera, an optical range finder and tactile sensors.

Limited perception capabilities and underdeveloped processing techniques were a common problem in early intelligent systems. Moreover, they worked only in highly structured indoor environments using specific landmarks or a limited set of obstacle shapes in order to clearly be recognized by the system. However, these works proved that even using limited hardware and software resources, it was still possible to artificiality observe, process and model the world surrounding the system, i.e. to perceive the world, an imperative prerequisite for autonomous navigation.

At the beginning of 1980, using vision techniques along with probabilistic approaches, Dickmanns was able to build cars that could run on empty roads autonomously. His works are regarded as the base of many modern approaches. By 1987, the intelligent vehicle *VaMoRs* developed by Dickmanns could run at speeds up to *96 km/h* and was a strong motivation for the development of the Prometheus project (1987-1995), proposed to conduct research for autonomous cars. This project produced two important

events: in 1994, a demonstration of two robot cars (*VaMP* and *VITA-2*) driving more than one thousand kilometres on a Paris multi-lane highway in standard heavy traffic at speeds up-to *130 km/h*; in 1995, an S-Class Mercedes-Benz car did a round trip from Munich to Copenhagen, driving up-to *175 km/h* on the German Autobahn (Dickmanns, 2007). Although the results were impressive and very promising for future research, these cars had many limitations. They could only drive using one lane of the road, this means lane changing and normal high-traffic manoeuvres were not possible. Obstacle detection was limited to vehicles in front of the car and human feedback was a constant input.

Development of new robotic techniques and hardware sensors provided more motivation for the development of intelligent systems for car applications. In 1998, Toyota cars became the first to introduce an Active Cruse Control (ACC) system on a production vehicle using a laser-based intelligent system. Daimler Chrysler developed a radar-based advance cruise control in 1999 for its high class cars and in 2000 it started to work on vision-based systems for lane departure manoeuvres for heavy trucks.

Later on, a series of DARPA grand challenges, developed by the U.S. government, fostered research and development in the area of autonomous driving. The challenge required an autonomous ground robot to traverse a *200 km* course through the Mojave desert in no more than 10 hours. While in 2004, no vehicle travelled more than 5% of the course, five vehicles finished the challenge in 2005. In 2007, the winners were able to complete another challenge by autonomously driving a *96 km* long urban track in less than 5 hours. These challenges represent a milestone in the autonomous driving field, however they were able to achieve these goals by using many redundant sensor configurations (Montemerlo *et al.*, 2006).

Recently, there are many research institutes or companies focusing on building autonomous cars. We can mention the driver-less cars from Google Labs[1] and from the Free University of Berlin[2] as modern systems that have set new milestones in autonomous vehicle driving. These cars use a sophisticated array of sensors for perception proposes, e.g., 3D lidar scanner, 2D lidar scanners, radar, mono or stereo cameras, among many software resources such that the total cost of sensors and add-ons is superior to the cost of the car itself.

Even with their impressive results, the intelligent vehicle systems mentioned above were not capable of dealing with all possible situations that we may encounter while driving in everyday scenarios such as urban areas, rural roads and highways. One of

[1] http://www.google.com/about/jobs/lifeatgoogle/self-driving-car-test-steve-mahan.html
[2] http://www.autonomos.inf.fu-berlin.de/

the main reasons is the detailed and precise information required about the environment. This information is delivered by the process of perceiving and processing the data coming from the sensors. One single sensor, due to its limitations and uncertainty, is not capable of providing all necessary information required by intelligent systems. Combining information from several sensors is a current state of the art solution and challenging problem (Vu, 2009).

Advance driver assistance systems (ADAS) are a particular kind of intelligent system. These systems help drivers to perform complex driving tasks to avoid dangerous situations. The main goal of an ADAS is to assist the driver in a safe navigation experience. Assistance tasks include: warning messages when dangerous driving situations are detected (e.g., possible collisions with cars, pedestrians or other obstacles of interest), activation of safety devices to mitigate imminent collisions, autonomous manoeuvres to avoid obstacles and attention-less driver warnings. There are three main scenarios of interest for ADAS: highway, countryside and urban areas. Each one of these presents different challenges: high speeds, few landmark indicators, and a cluttered environment with several types of objects of interest, such as pedestrians, cars, bikes, buses, trucks.

1.1 Perception for intelligent Vehicles

Environment perception is the base of an ADAS. This provides, by processing sensor measurements, information about the environment the vehicle is immersed in. *Perception* is the first of three main components of an intelligent system and aims at modelling the environment. *Reasoning & decision* uses the information obtained by perception to decide which actions are more adequate. Finally, the *control* component executes such actions. In order to obtain good reasoning and control we have to correctly model the surrounding environment. Figure 1.1 shows the interaction of the three main components of a generic intelligent system.

Perception consists of establishing the spatial and temporal relationships among the robot, static and moving objects in the scenario. As Wang summarizes in (Wang *et al.*, 2007), in order to perceive the environment, first, the vehicle must be able to localize itself in the scenario by establishing its spatial relationships with static objects. Secondly, it has to build a map of the environment by establishing the spatial relationships among static objects. Finally, it has to detect and track the moving objects by establishing the spatial and temporal relationships between moving objects and the vehicle, and between moving and static objects.

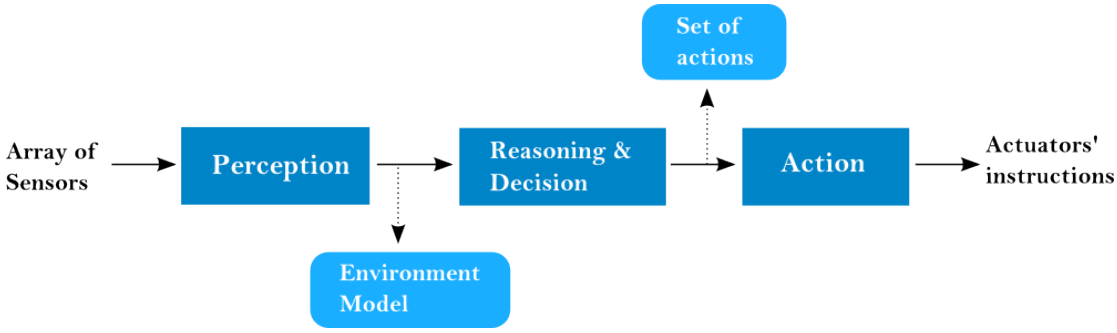


Figure 1.1: Architecture for an autonomous robotic system.

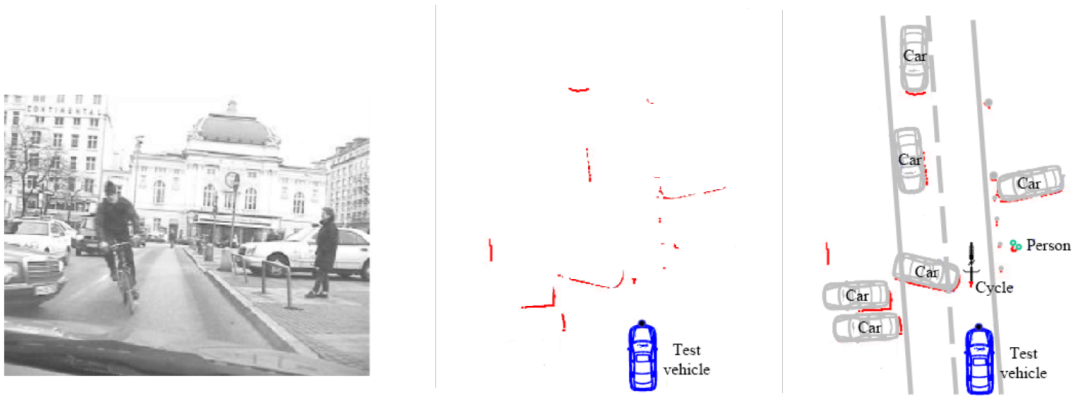


Figure 1.2: Example output of a perception system into a car. Left: image capture of a real urban scenario. Center: Raw measurements from a lidar sensor. Right: Environment representation of the real scenario, static and moving objects are highlighted.

Perceiving the environment involves the selection of different sensors to obtain a detailed description of the environment and an accurate identification of the objects of interest. Due to the uncertain and imprecise nature of the sensors measurements, all the processes involved in the perception task are affected, and have to manage uncertain inputs. Figure 1.2 shows an example of an environment representation for a real urban scenario, this figure shows the raw data provided by a lidar sensor and the final representation of the vehicle's surroundings obtained using this data. This representation should display the static and moving objects in the environment.

Vehicle perception is composed of two main tasks: simultaneous localization and mapping (SLAM) deals with modelling static parts; and detection and tracking of moving objects (DATMO) is responsible for modelling dynamic parts of the environment. In SLAM, when vehicle location and map are unknown the vehicle generates a map of the environment while simultaneously localizing itself within the map given all the measurements from its sensors. DATMO aims to detect and track the moving objects surrounding the vehicle and predicts their future behaviours. Figure 1.3 shows the

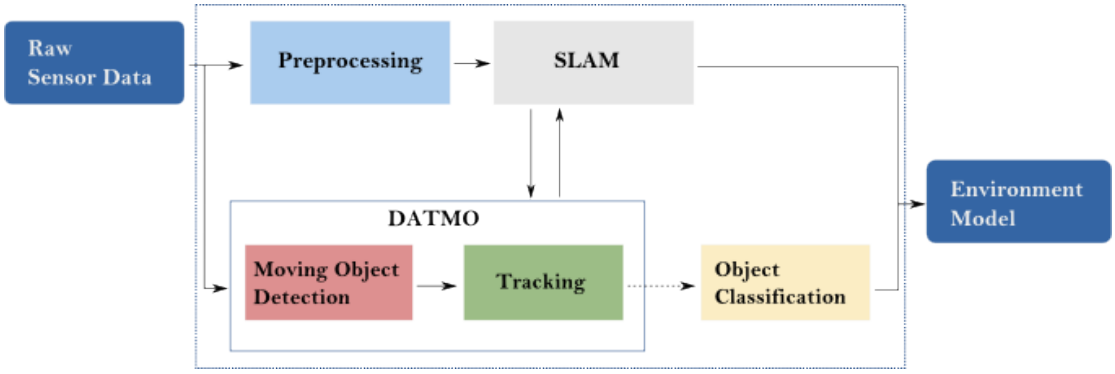


Figure 1.3: General architecture of the perception task and its two main components: simultaneous localization and mapping (SLAM) and detection and tracking of moving objects (DATMO).

main components of the perception task. It takes input sensor measurements and control inputs such as motion measurements, which after the preprocessing or refinement step become the input for SLAM. SLAM and DATMO are considered to be correlated by interactively sharing information (Vu, 2009, Wang *et al.*, 2007). The final output is a model of the environment usually composed by the ego-vehicle state, map of static components, and a list of moving objects and their description over time, e.g. position, velocity, type or class.

Management of incomplete information is an important requirement for perception systems. Incomplete information can be originated from sensor-related reasons, such as calibration issues, hardware malfunctions, miss detections, asynchronous scans; or from scene perturbations, like occlusions, weather issues and object shifting. It is important to manage these situations by taking into account the degree of imprecision and uncertainty.

If we are able to obtain reliable results from both SLAM and DATMO in real time, we can use the generated environment model to detect critical situations and warn the driver. Also, we can apply safety protocols automatically, like activating braking systems to avoid collisions, or if it is inevitable, to mitigate damage.

In the next sections we will briefly describe the main components of perception and narrow the domain of our contributions.

1.2 Preprocessing

A sensor is a device that measures and converts a physical magnitude into readable data that can be processed by a computer. Sensors are the input mechanisms of per-

ception systems and the only way to obtain direct information from the environment. Their counterparts are the actuators, which are the output mechanisms that interact with the environment. Usually, a sensor measures a single aspect of the environment; hence, to get information about all the needed aspects of the environment, different sensors are commonly used. This allows the ability to obtain a more accurate and complete environment representation. According to their interaction with the environment, sensors can be classified into two categories: passive and active. On one hand, passive sensors only measure the energy emitted by entities in the environment. On the other hand, active sensors emit energy into the environment and then observe the change this energy causes in the state of the environment.

When camera sensors are used to detect moving objects, appearance-based approaches are preferred and moving objects can be detected despite if they are moving or temporally static. If lidar sensors are used, feature approaches are usually preferred and relies on the moving behaviour of the obstacles through time (Wang and Thorpe, 2002).

The decision of which and how many sensors to use highly depends on the requirements of the application. For example, when high precision of the position of the objects of interest is needed, a 2D laser scanner is the most common choice, while when object recognition is required camera sensors could be better suited for the task. 3D laser scanners are becoming the most popular option nowadays due to their high precision and ability to extract accurate volumetric information of the objects from the scene. However, processing 3D data requires high computational resources. Besides, 3D laser scanners are not affordable for commercial applications.

1.2.1 Sensor uncertainty

In order to correctly model the behavior of sensors we have to be aware of their uncertain nature. Sensor uncertainty might come from hardware limitations which allow only to give an estimation of a real physical quantity. Environment noise or technical limitations and failures of the sensors can generate random errors in the sensor measurements, and represent another source of uncertainty. Uncertainty can come in a systemic way from calibration issues and from transformation errors from the sensor space to a common representation (Hall and Llinas, 1997, Mitchel, 2007).

Uncertainty management is an integral part within the perception task, it includes the association of uncertain and imprecise data with *partially complete* information, and the updating process of confidence factors. A sensor measurement is an approximation

of a real physical quantity. The confidence of a measurement represents a judgment about the validity of the information provided by this measurement. Its imprecision concerns the content of this information and is relative to the value of the measurement while the uncertainty is relative to confidence of the measurement (Dubois and Prade, 1985).

Sensor modeling is the description of the probabilistic relation between the sensor measurement and the actual state of the environment and is used to model its behavior (Elfes, 1989). Contrary to this concept exists the inverse sensor model which gives the probability of a certain environment configuration given a sensor measurement.

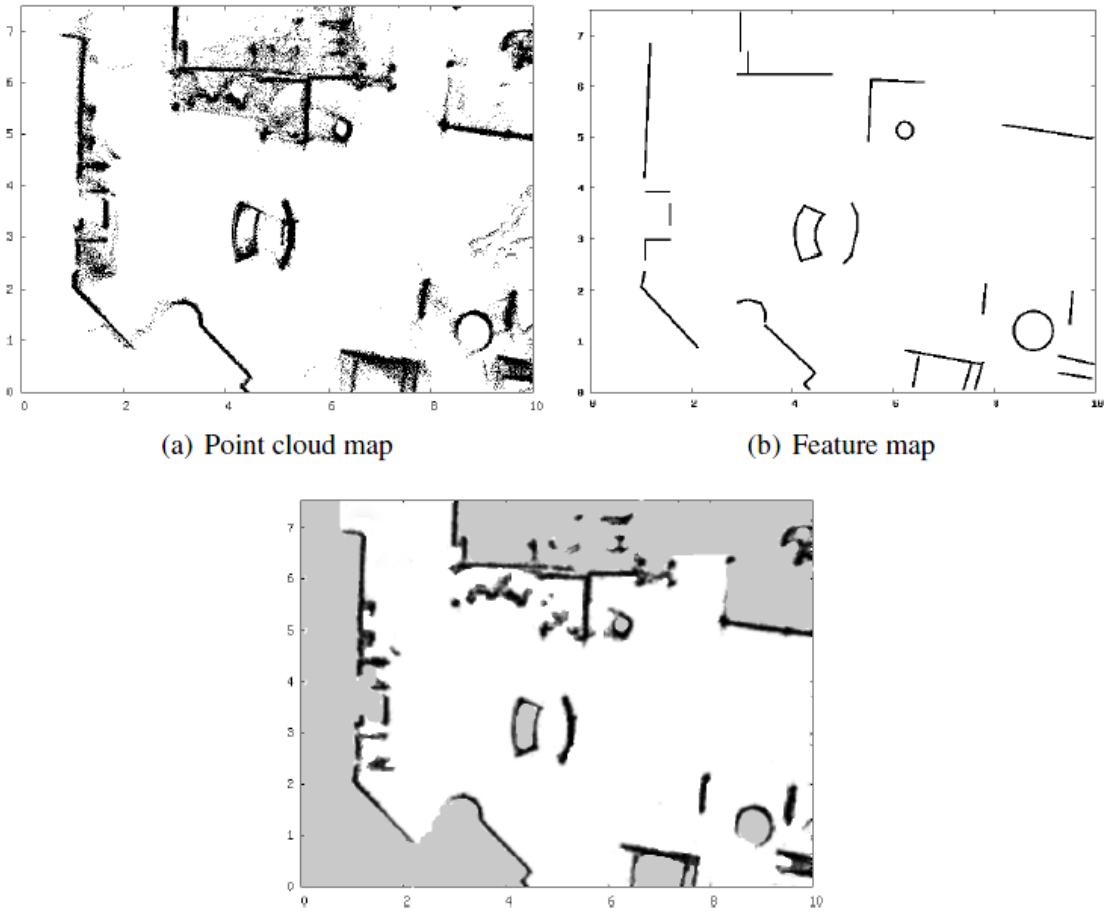
1.3 Simultaneous localization and mapping

Simultaneous localization and mapping (SLAM) allows robots to operate in an unknown environment and then incrementally build a map of the environment using information provided by their sensors; concurrently robots use this map to localize themselves. This map is supposed to be composed only of static elements of the environment. Moving objects have to be detected and filtered out using this so called *static map*.

In order to detect and track moving objects surrounding a moving vehicle, precise localization information is required. It is well known that position-based sensors like GPS or DGPS often fail in cluttered areas. Works like the one proposed by Wang *et al.* (2007), state that this issue is due to the *canyon effect*, and suggest (showing improvements regarding both tasks) a simultaneous relation between SLAM and DATMO.

Prior to detecting moving and static obstacles surrounding the vehicle, we must obtain a map representation of the environment. Three approaches are broadly used by current state of the art mapping applications:

- Direct approach uses raw data measurements to represent the physical environment without extracting predefined features, e.g. point clouds representations (Lu and Milios, 1997).
- Feature-based approach compresses measurement data into predefined features, for example geometric primitives like points, lines, circles (Leonard and Durrant-Whyte, 1991).
- Grid-based approach subdivides the environment into a regular grid of cells, and then a probabilistic or evidential measure of occupancy is estimated for each cell



(a) Point cloud map (b) Feature map

(c) Grid map

Figure 1.4: Examples of the three main approaches to perform map representation. The representation is based on lidar measurements.

(Elfes, 1989, Moras *et al.*, 2011b).

Despite the advantages of representing any kind of environment, direct approaches are infeasible regarding memory use and are unable to represent the sensors' uncertainty. Feature-based approaches are compact, but cannot properly represent complex environments using simple primitives. The grid-based approach is more suitable for our task because they are able to represent arbitrary features, provide detailed representations, and take into account sensor characteristics by defining sensor models. Figure 1.4 shows an example of the map representations obtained by the three main approaches mentioned above.

The occupancy grid approach has become the most common choice among map representation methods for outdoor environments due to its advantages over the others. Its main drawback is the amount of memory needed to represent a large outdoor environment; however, works like the ones presented by Wang and Thorpe (2002)

and Vu (2009) have proposed to overcome this issue, their idea is that since the sensor's range is limited, there is only a need to construct a local grid map limited by the non-measurable regions. Afterwards local maps are assembled to build a global map.

Vu (2009) proposes an interesting solution to solve the SLAM problem which is based on an occupancy grid to represent the vehicle map, and free-form objects to represent moving entities. To correct vehicle location from odometry he introduces a new fast incremental scan matching method that works reliably in dynamic outdoor environments. After a good vehicle location is estimated, the surrounding map is updated incrementally. The results of his method in real scenarios are promising. Therefore, we decided to use his method as a basis for our perception solution. However, our solution is not focused on the SLAM component but in the DATMO component of the perception problem.

1.4 Detection, Classification and Tracking of Moving Objects

Moving object detection aims at localizing the dynamic objects through different data frames obtained by the sensors in order to estimate their future state. The object's state has to be updated at each time instance. Moving object localization is not a simple task even with precise localization information. The main problem occurs when in cluttered or urban scenarios because of the wide variety of objects of interest surrounding the vehicle (Baig, 2012, Vu, 2009, Wang *et al.*, 2007). Two main approaches are used to detect moving objects. Appearance-based approaches are used by camera-based systems and can detect moving objects or temporally stationary objects. Featured-based approaches are used when lidar (laser scanners) sensors are present and rely on the detection of the moving object by their dynamic features. Both approaches rely on prior knowledge of the targets.

Although they become more necessary when in crowded areas, we can define three tasks independent of the scenario to attempt to solve the DATMO problem: detection of moving objects, moving object tracking (involving data association and motion modelling), and classification.

1.4.1 Detection of moving objects

Detection of moving objects focuses on discriminating moving parts from the static parts of the map. Due to the wide variety and number of targets that can appear in the environment and to their discriminative features, it is infeasible to rely on just one

environment representation to differentiate the moving objects from the static ones: e.g., feature or appearance-based representation. Whilst detecting moving objects, the system has to be aware of noisy measurements that lead to possible false detections of moving objects and mis-detections originated by the same noisy measurements or due to occlusions.

1.4.2 Tracking of moving objects

Once we detect the moving objects, it is necessary to track and predict their behaviour. This information is used afterwards in the decision-making step. Tracking of moving objects is the process of establishing the spatial and temporal relationship among moving objects, static obstacles and the ego-vehicle. The tracking process uses the motion model of the objects to properly estimate and correct their movement and position (Thrun *et al.*, 2005). The need of object tracking originates, in part, from the problems of moving object detection, for example, temporary occlusions or mis-detections of an object can be managed by the estimation of its position based on the object's motion model and motion history. Contrary to the SLAM problem, the DATMO problem does not have information about the moving objects' motion models and their odometry. Usually, this information is partially known and is updated over time; therefore, the motion models of moving objects have to be discovered on-line and the ADAS system has to be aware of the probable change in their dynamic behaviour. Ground moving object tracking presents a challenge due to the changes of motion behaviour of objects of interest. For example, a car can be parked or waiting at a traffic light and after moving at a constant acceleration (Chong *et al.*, 2000).

The basic scenario for a tracking task appears when just one object is being detected and tracked. Hence, in this basic scenario we can assume that each new object detection is associated with the single track we are following. However, this situation is far from the real scenarios where intelligent vehicles are deployed. When there are several moving objects, the tracking process becomes more complex and two main tasks emerge: *data association* and *track management*. The data association task is in charge of associating new object detections to existing object tracks whenever possible, whilst tracking management is in charge of maintaining the list of tracks up-to-date by applying operations over the tracks such as: creating tracks for new objects, deleting tracks of unobserved objects, and merging or splitting tracks.

1.4.3 Classification

Knowing the class of objects surrounding the ego-vehicle provides a better understanding of driving situations. Classification is in charge of tagging an object according to its type, e.g. car, pedestrian, truck, etc. We consider the class of the objects as an important addition to the perception process output but as well as key information to improve the whole DATMO component. Motion modelling provides preliminary information about the class of moving objects. However, by using appearance-based information we can complement knowledge about the class of objects the intelligent vehicle is dealing with despite the temporal static nature of the objects or partial misdetections. Although it is seen as a separate task within the DATMO task, classification can help to enrich the detection stage by including information from different sensor views of the environment, e.g. impact points provided by lidar, image patches provided by camera. Evidence about the class of objects can provide hints to discriminate, confirm, or question data association decisions. Moreover, knowing the class of a moving object benefits the motion model learning and tracking which reduces the possible motion models to a class-related set.

An intelligent vehicle can be positioned in several kinds of scenarios. These scenarios can contain many kinds of moving objects of interest, such as pedestrians, bicycles, motorcycles, cars, buses and trucks. All these objects can have different appearances, e.g. geometry, textures, intra-class features; and different dynamic behaviour, e.g. velocity, acceleration, angular variation. When using laser scanners, the features of moving objects can change significantly from scan to scan. When using camera sensors, the quality of visual features can be drastically decreased due to occlusions or weather conditions. As a result, it is very difficult to define features or appearances for detecting specific objects using laser scanners. Moreover, a single sensor is not enough to cover all the objects of interest in a common driving situation. For several decades, the need for assembling arrays of sensors for intelligent vehicle perception has been a requirement to provide robustness for ADAS.

Most of the current DATMO solutions consider classification information only as an aggregate information for the final perception output. Few considerations have been made to include object class information as complementary data for object detection, and use it as a discrimination factor for data association and as feedback information for moving object tracking. We believe that classification information about objects of interest gathered from different sensors can improve their detection and tracking, by reducing false positive detections and mis-classifications. Moreover, a fusion solution for multiple sensor inputs can improve the final object state description.

1.5 Simultaneous localization, mapping and moving object tracking

Both SLAM and DATMO have been studied in isolation (Bar-Shalom and Li, 1998, Blackman, 2004, Civera *et al.*, 2008). However, when driving in highly dynamic environments, such as crowded urban environments composed of stationary and moving objects, neither of them is sufficient. Works like the ones reported by Wang *et al.* (2007) and Vu (2009) proposed a solution to the problem of simultaneous localization, mapping and moving object tracking which aims at solving the SLAM and the DATMO problems at once. Because SLAM provides more accurate pose estimates and a surrounding map, a wide variety of moving objects are detected using the surrounding map without using any predefined features or appearances, and tracking is performed reliably with accurate vehicle pose estimates. SLAM can be more accurate because moving objects are filtered out of the SLAM process thanks to the moving object location prediction from DATMO. SLAM and DATMO are mutually beneficial.

Vu (2009) proposes a perception solution based on the simultaneous relation between SLAM and DATMO. Although he considers classification information, he uses it as an aggregation to the final output and not as interactive factor to improve the perception result. Moreover he uses the class information in a deterministic way which disables the uncertain management and indirectly limits the real-time capabilities of his solution. Although focusing on the DATMO component, our perception solution includes the class information as a distribution of evidence considering uncertainty from the sources of information and transferring this evidence until the final result.

1.6 Multi-sensor fusion for vehicle perception

Although combining information from different senses is a quite natural and effortless process for human beings, to imitate the same process in robotics systems is an extremely challenging task. Fortunately this task is becoming viable due to the availability of new sensors, advanced processing techniques and algorithms, and improved computer hardware. Recent advances in computing and sensing have provided the capability to emulate and improve the natural perceiving skills and data fusion capabilities.

Information fusion consists of combining information inputs from different sources in order to get a more precise (combined) output which is generally used for decision-making tasks. This process includes three main subtasks: data association, which finds

the data correspondence between the different information sources; estimation, combines the associated data into an instantaneous value; and updating, which incorporates data from new instances into the already combined information. Improvements in fusion approaches can be made by focusing on the data representation for individual and combined values, the design architecture of the fusion approach, and in the management of uncertain and imprecise data.

Sensors are designed to provide specific data extracted from the environment. For example, lidar provides features like position and shape (lines) of obstacles within its field of view, camera sensors provide visual features that can be used to infer appearance information from obstacles, like class of the objects. Intelligently combining these features from sensors may give a complete view of the objects and the environment around the intelligent vehicle and it is the objective of a fusion architecture. Multi-sensor fusion has become a necessary approach to overcome the disadvantages of single-sensor perception architecture and to improve the static and dynamic object identification providing a richer description. Improvements in sensor, processor and actuator technologies combined with the trending initiative of the automotive industry and research works have allowed the construction and application of several intelligent systems embedded in modern consumer vehicles.

Real driving scenarios represent different challenges for the perception task: high speeds, few landmark indicators, and a cluttered environment with different types of objects of interest. Several research works have proposed that using information from several sensors may overcome these challenges by obtaining redundant or complementary data. For example, camera sensors can be used to capture the appearance of certain objects or landmarks using features like colors, textures or shapes. Camera data is sensible to frequent changes in the ambiance. Other sensors, like lidar, can provide complementary or redundant information to deliver a set of discrete points of impact on static and moving objects. It is clear that there is no universal sensor hardware capable of sensing all the measurements that an intelligent vehicle needs to precisely perceive every environment. For this reason, we decide to include many sensors as the interface with the environment to complement or verify the information provided by them. However optimally fusing information from these sensors is still a challenge.

Fusion methods applied to the vehicle perception can help to: enrich the representation of the surrounding environment using complementary information; improve the precision of the measurements using redundant information from different sensors or other dynamic evidence sources; and manage uncertainty during the fusion process by identifying and associating imprecise and uncertain data which would help to update

the confidence levels of fused information. Fusion approaches have to take into account the nature of the data provided for each sensor, sensors' field of view, degree of redundancy or complementation of provided data, synchronization of the inputs, and the frequency of the fusion mechanism.

Although information fusion from different sensors is a challenging task for autonomous and even semi-autonomous vehicles, due to new sensor technologies and improvements in data processing algorithms this task is becoming more feasible. Modern computer techniques and hardware have helped current intelligent vehicles systems to fuse information from a wide range of sensor types, e.g. passive sensors like single and stereo cameras, or active sensors like radar or lidar.

Several methods have been proposed to combine information from different sensor observations, these can generally be classified by the approach they are based on: Bayesian, Evidential and fuzzy. Most of them are thought to work with raw data, but when data at detection level is available similar approaches can be used (Castellanos *et al.*, 2001). Methods that use raw information as inputs (e.g. lidar scans) have to perform the very first stages of the perception process: map building, localization and moving object detection. Sometimes modern sensors just provide information at detection level, i.e. a list of detected objects, in this case the fusion process is expected to provide a unique list of objects to improve the individual ones. Our work focuses on fusing objects' state information at two levels within the DATMO problem. The fusion approaches take into account all the available information from observed and known objects, such as kinetic information (position, velocity) and class information. The purpose of these approaches is to observe the degree of improvement achieved by the inclusion of class information at the detection level and at the tracking level.

Although there are several approaches in the current literature trying to improve the perception output by fusing information from different sensors, few of them consider the uncertainty and imprecision from sensors and integrate them into the detection and tracking stages of DATMO task. The transferred belief model (TBM) represents, in a quantified way, beliefs obtained from belief functions. This model is able to explicitly represent uncertainty and inaccuracy of an event. Also, it manages quantitative and qualitative data and models the unknown evidence as well as the known evidence. Within the intelligent vehicle perception, TBM has been considered as the main technique for sensor fusion, scene interpretation, object detection, object verification, data association and tracking. This model proposes an alternative to Bayesian theory and overcomes some of its limitations by being able to manage uncertainty, transfer belief measures, propose inference mechanisms and to be clearly analogous to human

reasoning process. Our proposed work relies on TBM to represent object class hypotheses, perform evidence fusion information and to deal with data association problem at different levels in the perception process.

1.6.1 Requirements of multi-sensor systems

In previous sections we have introduced the problem of single-sensor perception systems and the idea of how multiple sensors can be included inside the perception task. In order to develop a multi-sensor system, we have to propose and implement several modules. First, sensor data processing takes the raw data from sensors and using sensor-based models, transforms this raw data into useful information. Raw data processing modules depend on the number and type of sensors installed on the demonstrator. Second, SLAM modules take the processed data to create a map and perform localization. These modules might need to implement data fusion at this stage. Third, preprocessed data and the output from the SLAM component are taken by the DATMO modules to identify, classify and track moving objects. Fusion at this level combines lists of tracks, and in case no data fusion was performed in SLAM component, it must implement fusion of detected objects.

In this dissertation, we propose two multi-sensor fusion approaches at DATMO level. Therefore, we follow the data flow described above. In Chapter 2, after the state-of-the-art description, we introduce the data processing modules for lidar, radar and camera which are common to both proposed fusion approaches. Besides, in Chapters 4 and 5 we describe in detail our different proposed modules inside the two fusion approaches and how the data processing modules take place to construct a whole perception solution.

1.7 Contributions

The approach proposed in this dissertation focuses on the DATMO stage of the perception problem. Regarding the state of the art approaches, we assume the SLAM stage as a solved task, and focus on the detection, tracking and classification of moving objects. Precisely, we believe that including objects class as a key information along managing the uncertainty from sensor data processing we can improve the results of the perception task, i.e. a more reliable list of moving objects of interest represented by their dynamic state and appearance information. Therefore, we address the problems of sensor data association, sensor fusion for object detection and tracking at different

levels within the DATMO stage. We assume that a richer list of tracked objects can improve future stages of an ADAS and enhance its final application.

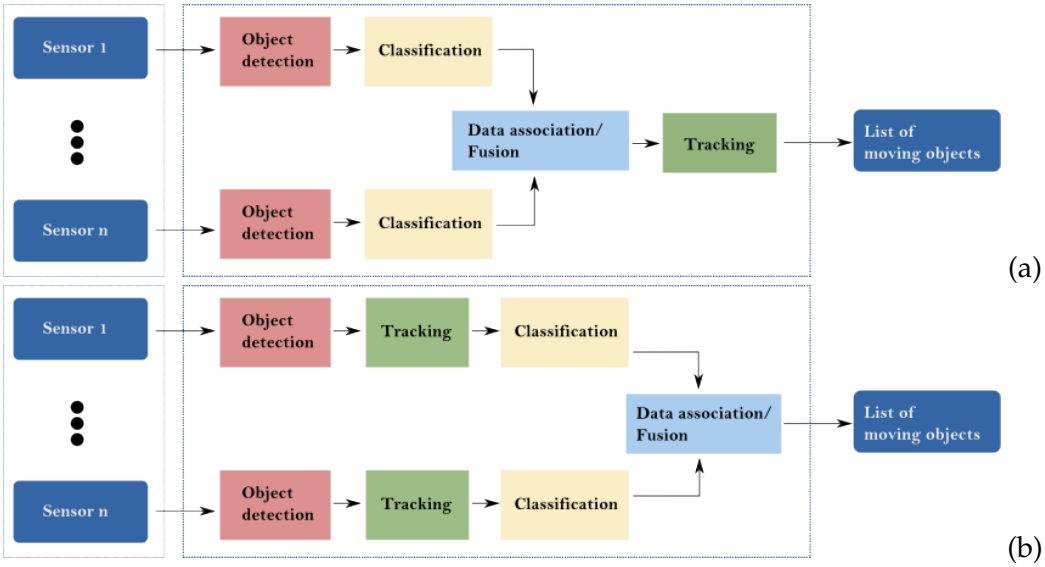


Figure 1.5: Architecture of the proposed approaches for data association and fusion at two different levels: (a) detection level, (b) tracking level.

Before focusing on our proposed approaches we reviewed the state-of-the-art works regarding the perception task for intelligent vehicle systems, focusing on the automotive applications for outdoor environments. This review considers not only the DATMO but the SLAM task to be aware of the closer relation and interaction between them. We follow works regarding sensor data processing, environment representation, moving object dynamics modelling, data association, filtering and fusion techniques. Although we focus on a set of three main sensors: radar, lidar and camera, we propose a modifiable architecture to include other types or number of sensors. A detailed description of the set of sensors and the vehicle architecture is given in Section 3.5. Even though we are describing our contributions in detail later in this dissertation, we can briefly mention them as follows.

First, we define an object representation to include class information as a part of the object state from early stages to the final output of the perception task and not only as an aggregate output value.

Second, we propose, implement and compare two different perception architectures to solve DATMO problem according to the level where object association, fusion and classification information is included and performed. Our data fusion approaches are based on the evidential framework and are composed of instantaneous fusion of current measurements and dynamic fusion of previous object states and current evidence. Figure 1.5 shows the architecture of the two proposed approaches.

Third, we propose an evidential data association approach to establish a relationship between two sources of evidence from object detections. We apply this approach at the detection level to fuse information from two sensor detections and at tracking level to find the relations between observations and tracks of known objects. We observe how the class information improves the final result of the DATMO component.

Fourth, we integrate the proposed fusion approaches as a part of a real-time vehicle application. This integration has been performed in the framework of the *interactIVe* European project. This project pursues the development of an innovative model and platform for enhancing the perception of the traffic situation in the vicinity of the vehicle. Precisely, our work takes place in the perception stage and integrates different information sources like raw sensor data, state of the vehicle, and road edge detections as inputs to perform the perception task. Moreover, our implementation provides a list of tracks of moving objects of interest and identifies the static obstacles in the environment. Given the presence of different sources of evidence and uncertainty, we tested our two fusion strategies to include classification information as a key parameter to improve the perception output.

Fifth, we analyse and experimentally evaluate the performance of the proposed methods based on the detection, tracking and classification of multiple objects in different driving scenarios. We compare our evidential approaches against each other and against a state-of-the-art work proposed by Vu (2009) using real data from a vehicle demonstrator of the *interactIVe* project. This comparison focuses on the detection and classification of moving objects.

1.8 Thesis outline

The organization of the rest of this dissertation is as follows. In Chapter 2, we review the state-of-the-art works that solve the perception problem. We start by formally introducing the two main components of the intelligent vehicle perception: SLAM and DATMO. Then, we focus on the DATMO component analysing the different work proposals for each of its internal subtasks. After analysing the single sensor problem, we review the different methodologies and architectures for multi-sensor solutions that are related to our contributions.

Chapter 3 presents the sensor configuration we use in this dissertation and the sensor processing methods implemented to gather, process, and represent the environment. The methods presented in this chapter are common to our two multi-sensor contributions. Also, this chapter presents an overview of the *interactIVe* European project

highlighting the goals and requirements of the perception system inside. This overview helps us to visualize the structure of the real perception system we will detail in Chapter 6. In order to give a general perspective of our two fusion architectures, in Chapter 3, we also introduce an overview of the general architectures that will be detailed in Chapters 4 and 5.

In Chapter 4, we describe the first complete DATMO solution using our fusion approach at tracking level. Here, we follow the common late fusion architecture adding class information to the object representation and using it to improve the final moving object classification. Two main trackers are implemented using lidar and radar data. Preliminary class information is extracted from the three sensors after moving object tracking is performed. Experimental results are presented to verify the improvement of the fusion approach with respect to the single-sensor trackers.

In Chapter 5, we describe a second complete DATMO solution using our fusion approach at detection level. Here, we proposed several classification methods based on the early object detections from the three sensors. We propose and build a composite object representation at detection level based on the kinematic and class information of the detected objects. We also propose a data association method to find relations between different lists of moving object detections. Besides, an enhanced version of the moving object tracking presented in Chapter 4 is described using the aforementioned composite representation. We conduct several experiments to compare the two proposed fusion approaches and analyse the improvement of an early consideration of the class information inside the DATMO component.

The description of a real intelligent vehicle application using our composite object description proposal and a combination of the methodologies inside our fusion approaches is detailed in Chapter 6. This application is built inside the *interactIVe* European project and tested in on-line and off-line demonstrations. In the same chapter, an experimental set-up, several tests in real driving scenarios and an evaluation section are included to verify the performance of our contributions inside the whole intelligent vehicle solution. Finally, we summarize our conclusions and present some perspectives of future works in Chapter 7.

CHAPTER 2

Intelligent Vehicle Perception

PERCEPTION consists of establishing the spatial and temporal relations between the vehicle and its surrounding environment. In this chapter, we review the state-of-the-art of the main components of vehicle perception problem: SLAM and DATMO. First, we formally present the problem of perception for intelligent vehicle systems. Second, we define the component of SLAM and focus on DATMO component where our contributions are present. Third, we analyse separately the modules inside DATMO to present the main related works. Fourth, we present the problem of multi-sensor fusion and offer a review of the state-of-the-art according to the most used fusion techniques and to the different fusion architectures. Finally, we summarize the analysis of the state-of-the-art and highlight the fusion methods and fusion architectures present in our contributions.

2.1 Vehicle perception problem

In Chapter 1, we presented the problem of perception for intelligent vehicles as the process of generating a representation of the environment surrounding the vehicle by interpreting the data from vehicle sensors. This process involves the estimation of vehicle position in the environment, the relative positions of static and moving objects around the vehicle, and the tracking of objects of interest. We can see a graphical representation of the perception problem in Figure 2.1. In order to discuss state-of-the-art approaches that have been developed to solve the perception problem, we need to formally define this problem. Given a set of sensor observations Z_t defined as:

$$Z_t = \{z_0, z_1, ..., z_t\}, \tag{2.1.1}$$

and control inputs U_t up to time t defined as:

$$U_t = \{u_1, u_2, ..., u_t\}, \tag{2.1.2}$$

the perception process is divided in two main components: simultaneous localization and mapping (SLAM) which captures the static part of the environment; and detection and tracking of moving objects (DATMO) which determines and follows the moving entities over time. These components are defined in detail in Sections 2.2 and 2.3 respectively.

Figure 2.1: Process diagram for intelligent vehicle perception.

Vehicle perception provides, up to time t, three main outputs: first, the set of estimated vehicle states:

$$X_t = \{x_0, x_1, ..., x_t\}, \tag{2.1.3}$$

usually defined by its position and orientation; second, the set of i static objects in the environment, also known as the map:

$$M_t = \{m_0, m_1, ..., m_K\}, \tag{2.1.4}$$

where K is the total number of objects in the environment, and each m_k, with $1 \leq k \leq K$, specifies properties and location of each object; finally, the set of J moving object tracks at time t, for $1 \leq j \leq J$ is defined as:

$$T_t = \{o_1, o_2, ..., o_J\}. \tag{2.1.5}$$

Current literature show that we can define the perception problem as an *a posteriori* probability calculation:

$$P(x_t, M_t, T_t | Z_t, U_t, x_0), \tag{2.1.6}$$

where x_0 represents the initial state of the vehicle. Moreover, as was mentioned above, we can divide the perception into SLAM:

$$P(x_t, M_t | Z_t^s, U_t, x_0), \tag{2.1.7}$$

and DATMO:

$$P(T_t | Z_t^d, x_t, M_t), \tag{2.1.8}$$

where Z_t is decomposed into static Z_t^s and dynamic observations Z_t^d:

$$Z_t = Z_t^s + Z_t^d. \tag{2.1.9}$$

Usually, state-of-the-art approaches consider SLAM and DATMO as separate problems. However, recent research works have proposed several methods that perform SLAM and DATMO simultaneously. This general problem is known as Simultaneous Localization, Mapping and Moving Object Tracking (SLAMMOT). Several works, such as the ones presented by Hähnel *et al.* (2003) and Montesano *et al.* (2005), have proposed simplified SLAMMOT solutions for indoor environments. Fortunately, recently works like (Wang *et al.*, 2007) and (Vu, 2009) have developed real-time applications for outdoor environments which represent the milestone we follow in this dissertation.

In the following sections we describe in detail SLAM and DATMO components; and their interaction as a correlated problem. We present the single sensor solution to solve the perception problem and afterwards we introduce the fusion solutions where our proposed approaches take place. Besides, we present the state-of-the-art approaches focusing on the place their main ideas take part. This allows us to properly position our proposed works and contributions.

2.2 Simultaneous Localization and Mapping

Knowing the current position or the localization of an autonomous robot is a key part of robotic perception. Imagine we put a robot inside an unknown environment. The robot has to be able, by using sensors' inputs, to generate a map of the environment, but to do so, it has to be aware of its localization inside this unknown environment, and vice versa. The problem becomes challenging due to the paradox inside it: to move precisely, a robot must have an accurate map of the environment; however, to build an accurate map, the robot needs to know its precise location in the map. The strong dependence between these two problems make SLAM one of the hardest problems in robotics. Simultaneous Localization and Mapping (SLAM) is a restatement of the previous problems: a robot placed at an unknown position in an unknown environment has to be able to *incrementally* build a consistent map of the environment, by using sensor inputs, while *simultaneously* determine its location inside the map. In summary, in a SLAM solution the robot acquires a map of the environment while simultaneously localizing itself within this map given all measurements from odometry and input sensors (e.g. camera, radar, lidar).

Initial solutions to solve SLAM can be traced back to the works proposed by Smith *et al.* (1987) and Leonard and Durrant-Whyte (1991). In these works, the authors establish a statistical basis for describing uncertain relationships between landmarks by manipulating geometric uncertainty and measurement uncertainty. Besides, these works discovered that estimates of landmarks observed by the vehicle are correlated because of the common error in vehicle location estimation. Therefore, positions of all landmarks were made part of the state vector and were updated on each observation. Computational and storage needs increased while the robot explored the unknown environment looking for new landmarks. In order to limit the resources explosion, many solutions establish thresholds or do not consider correlations between landmarks (Leonard and Durrant-Whyte, 1991, Thrun, 2000). These works are considered as the basis of modern SLAM solutions.

Landmarks were the main feature used to represent the environment and powered initial SLAM solutions. However, relying on landmarks to discover a dynamic outdoor environment proved not to be enough. Elfes (1989) introduced the occupancy grids (OG) framework for localization and mapping. In this framework the real environment is divided (discretized) into cells where each cell has a probabilistic estimate of its occupancy state. Usually a high value of cell probability indicates the cell is occupied and a low value means the cell is free or empty. Occupancy values are constantly updated with new sensor measures by applying Bayesian methods. It is clear that the computational and storage resources are directly related with the grid resolution. However, several approaches have shown that it is possible to obtain a trade-off between representation power and time processing for real time applications (Baig, 2012, Vu, 2009, Wang *et al.*, 2007). Works like the ones proposed by Moras *et al.* (2011a) and Pagac *et al.* (1998) show that occupancy can be represented using Dempster-Shafer theory considering evidence distributions that are able to represent one occupancy state without making assumption about the others. These works offer an interesting solution to make the difference between unknown (no information) and doubt caused by conflicting information gathered incrementally about the cell occupancy. Moreover, occupancy grid approaches establish a common representation which can be used to perform sensor fusion.

A highly used solution for SLAM is the probabilistic SLAM proposed by Vu (2009) and Wang and Thorpe (2002). This solution is based on Bayes' rule and Markovian assumptions, and requires the estimation of the following probability distribution:

$$P(x_t, M_t|z_{0:t}, u_{0:t}) \propto P(z_t|x_t, M_t) \int P(x_t|x_t - 1, u_t)P(x_t - 1, M_t|z_{o:t-1}, u_{1:t-1})dx_{t-1}, \tag{2.2.1}$$

which is composed of two recursive steps: prediction and correction.

The prediction step performs an updated based on time:

$$P(x_t, M_t|z_{0:t}, u_{1:t}) = \int P(x_t|x_{t-1}, u_t)P(x_{t-1}, M_t|z_{0:t-1}, u_{0:t-1})dx_{t-1}. \tag{2.2.2}$$

The correction step updates the information based on the new available measurements:

$$P(x_t, M_t|z_{0:1}, u_{1:t}) = \frac{P(z_t|x_t, M_t)P(x_t, M_t|z_{0:t}, u_{0:t})}{P(z_t|z_{0:t-1}, u_{0:t})}, \tag{2.2.3}$$

where the factor $P(x_t|x_{t-1}, u_t)$ is known as the vehicle model and $P(z_t|x_t, M_t)$ as the sensor model.

The vehicle model indicates how the vehicle has moved from its previous to current state according to the control inputs (e.g. control commands or odometry values), while the sensor model defines the probabilistic description of the sensors used to perceive the environment.

2.2.1 Map representation

Choosing how to represent the map of the environment is a very important step and the basis for future perception tasks. This decision depends on the application and relies on four main factors: data compression, level of environment representation, uncertainty management and sensor parameters. Following these factors and the classification proposed by Vu (2009), the most popular representation approaches are:

- Direct representation. Raw sensor data is used to represent the map. This method is also known as *point cloud* due to the use of points (mainly range sensors) as the main particles for representation. It is a straightforward method with a high level of representation, but lacks uncertainty management and its computational cost is high. Lu and Milios (1994) and Cole and Newman (2006) employ this representation for mapping by using range sensor scans, and for SLAM in 3D environments, respectively.

Table 2.1: Feature comparison among the main map representations schemes.

Scheme	Data compression	Detailed representation	Uncertainty management	Sensor features
Direct		X		
Feature-based	X		X	
Grid-based		X	X	X

- Feature representation. Raw sensor data is processed to identify and extract specific features which are used to represent the map. Using features provides a more compact representation. The level of representation highly depends on the set and number of features. Geometrical features are preferred but are less effective in irregular scenarios where the diversity of shapes become a challenge. Leonard and Durrant-Whyte (1991) and Dissanayake *et al.* (2001) proposed SLAM solutions using a set of geometrical shapes and landmarks to represent the environment.

- Grid-based representation. Originally proposed by Elfes (1989), in this representation the environment is discretized into regular cells and the occupancy state of the cell is inferred from the sensor measures at each time. This is a popular approach for indoor and outdoor perception applications. The level of representation and data compression is highly dependent on the grid resolution.

We consider grid-based representation as the best suited for our proposed approach. Its advantages, compared to the other representations schemes, can be seen in Table 2.1. Although grid-based representation may require high amounts of storage (lack of data compression) and computing processing, it is able to represent arbitrary features and provide detailed representations of the environment. Besides, works over grid-based representation have shown that its implementation is straight forward. Additionally, it has the ability to manage uncertainty from sensor measures and include sensor features.

2.3 Detection and Tracking of Moving Objects

Dynamic environments have many static and moving objects interacting with each other. Distinguishing moving objects is an important part of the perception task. Detection and tracking of moving objects are usually considered as two separated tasks

and performed as so, but they are highly related and can be seen as whole. For this reason both tasks are the core of the second part of the perception problem: detection and tracking of moving objects (DATMO). This problem has been widely studied and several approaches have been proposed. Even though the first solutions were proposed for military appliances (Singer and Stein, 1971), such as surveillance systems or embedded in sea and sky vehicles (Bar-Shalom and Li, 1998, Bar-Shalom and Tse, 1975, Fortmann *et al.*, 1983), many modern solutions are focused on the automotive industry and on service robotics (Baig *et al.*, 2011, Burlet *et al.*, 2006, Chavez-Garcia *et al.*, 2012, Wang *et al.*, 2007).

Many object tracking approaches rely on the correct detection of moving objects and therefore focus on the measurements-track association, tracks management, and in tracking itself. However, moving object detection does not provide a false-positive free list of moving objects. It is clear that, in order to perform a correct tracking, the moving object detection is a critical aspect in a DATMO system.

Usually, DATMO is divided into three main subtasks: detection of moving objects, moving object tracking, and data association. In the next sub sections we will describe in detail these subtasks and show the state-of-the-art solutions.

2.3.1 Moving Object Detection

Moving object detection is strongly related with the environment mapping task. The general idea focuses on identifying the dynamic part of the environment once the static part has been identified. The dynamic part contains the moving object entities of the environment.

As in mapping, there have been a vast group of research works on moving object detection. We can mention the different approaches according to the main sensor inputs used to identify moving objects.

The computer vision community has proposed several works in this area. The first approaches were based on the background subtraction technique to recognize moving patches (KaewTraKulPong and Bowden, 2002, Li, 2000, Ramesh *et al.*, 1979, Stauffer and Grimson, 1999). Unfortunately, dynamic environments represent a challenge to background subtraction solutions. Detection based on visual features techniques proposed the recognition and tracking of specific visual features that belong to moving objects (Tomasi and Kanade, 1991). Recently, modern techniques have assembled solutions using machine learning tools to build classifiers and recognized moving or potential moving objects in a camera image (Chavez-Garcia *et al.*, 2013, Yilmaz *et al.*,

2006).

Other approaches use information from ranged sensors to build a representation of the map and highlight inconsistencies, assuming these belong to moving objects. The occupancy approach is the most used representation. Using the map constructed by SLAM, these approaches analyse the new observations and, focusing on the changes, determine if a part of the map is moving or static. If moving parts are found, a cluster process is performed identifying individual moving objects (Garcia *et al.*, 2008, Vu, 2009, Wang and Thorpe, 2002). Although these techniques use probabilistic grids to represent the occupancy of the map, recent works are proposing evidential grids based on Dempster-Shafer Theory to represent and update the occupancy state of the grid using belief functions (Chavez-Garcia *et al.*, 2012, Moras *et al.*, 2011a,b). Recently, works like the one detailed by Vu (2009) have proposed model based techniques to detect moving objects on the roads in the context of autonomous vehicle driving.

As stated earlier in this chapter, we have decided to use the occupancy grid approach to represent the environment. Therefore, we use this map representation to discover the moving objects. Following the work of Vu (2009), we perform an occupancy grid solution based on lidar data where, $M_t[z_t^i]$ represents the cell of the map M_t, z_t^i represents the ith part (beam) of the measurement z (laser scan) at time t. Therefore, we define the dynamic parts of the measurements as the set $\{z_t^i | M[z_t^i] \text{ is marked empty}\}$.

2.3.2 Tracking of Moving Objects

Once moving observations are separated from static entities, those observations become the input for a moving object tracking module. Object tracking allows the aggregation of object observations over time in order to enhance the estimation of their dynamic states. The state vector can include position, speed, acceleration, geometric description and classification information. Usually, these state variables cannot be observed or measured directly, but they can be estimated through a tracking process. We can describe a track as a succession of object's states over the time. Despite the fact that there are several proposed solutions to perform the tracking of moving objects, those works follow a common structure (Thrun, 2000, Wang *et al.*, 2007). Figure 2.2 shows the common architecture of a moving object tracking (MOT) module. Once the MOT module receives the list of moving observations it performs a gating process to localize an area around the next predicted position of a track, this allows it to focus on an area where a new observation of a track is expected to appear. Next, a data association is performed between the tracks and the observations inside their gate areas, it is important to mention that this association is exclusive. Track management provides and

executes rules to keep an updated list of tracks, e.g., creation, deletion, confirmation, merging, splitting. Filtering is the recursive process of estimating the state of the tracks in the list provided by the track management, these estimations are used to perform the gating in the next time $(t+1)$. The moving object tracking module delivers a list of tracks of moving objects present at current time.

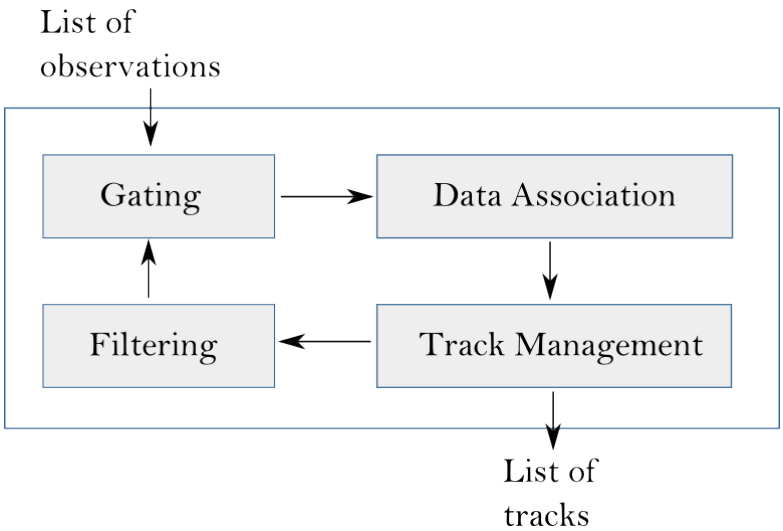


Figure 2.2: Common architecture of a Moving Object Tracking module.

Several MOT solutions have been proposed since the early days of research over this topic (Bar-Shalom and Tse, 1975, Reid, 1979, Singer and Stein, 1971). Even though these works did not make the clear separation of modules we show in Figure 2.2, in their implementations they differentiate the filtering from the data association stage. Usually, filtering inherently contained gating, while data association was in charge of managing the detected tracks. Hence, we can sort the related works based on their filtering and data association approaches.

Following the Bayesian approach we can define the MOT problem as follows. Consider moving object detection and tracking in a sliding window of time $[1, T]$. Let Z be the set of all data measurements within the time interval $[1, T]$ and $Z = \{z_1, ..., z_T\}$ where z_t denotes sensor measurement at time t. Assuming that within $[1, T]$ there are K unknown number of moving objects. The moving object detection and tracking problem is formulated by maximizing a posterior probability (MAP) of an interpretation of tracks of moving objects w for a set of sensor measurements Z:

$$w = \arg\max_{w \in \Omega} P(w|Z), \tag{2.3.1}$$

where Ω is the set of all possible track hypotheses for the K objects. If we apply Bayes

rule, Equation 2.3.1 can be decomposed into the prior model and the likelihood model:

$$P(w|Z) \propto P(w)P(Z|w). \tag{2.3.2}$$

2.3.3 Data Association

The data association task originated from the multi-object tracking problem. It can be defined as the task of associating which observation, from a group of observations given by a sensor processing module, was originated by which object. As mentioned in Section 1.2, there is uncertainty in sensor measurements that might generate unwanted observations. Moreover, the number of observations might not correspond with the number of objects, due not just to measurement issues that can add false or ambiguous observations, but as well temporary occlusions or objects out of the sensing area. All of these situations add more difficulty to the already challenging data association task.

According to the association technique we can divide the state-of-the-art data association approaches as follows.

Global Nearest Neighbor (GNN) was the first data association approach, which attempts to find and to propagate the single most likely hypothesis at each data sensor scan (Baig, 2012, Blackman, 2004). The term global comes from the fact that the association assignment is made considering all possibilities within a gate or area of association. It follows the constraint that an observation can be associated with at most one known object (or track). GNN approaches work well if the sensor data is not very noisy and the observations and known objects are sufficiently apart from each other. Its main drawbacks come from its application in cluttered scenarios, such as urban areas; and for relying on one-frame associations (Blackman, 2004).

Probabilistic Data Association combines all the potential candidates for associations to one track into a single statistic model, taking into account the statistical distribution of the track errors and clutter. This technique assumes that only one of the candidates is a valid observation, taking the rest as false alarms. Originally proposed by Bar-Shalom and Tse (1975), this technique has become the basis of many modern probabilistic based data association approaches, such as the Joint Probabilistic Data Association (Bar-Shalom and Li, 1998, Bar-Shalom and Tse, 1975).

Evidential Data Association proposes a similar background idea as the probabilistic data association. However, instead of using probabilistic distributions to represent the association relations, it uses the Transfered Belief Model (TBM) to represent the possible association hypotheses. By interpreting belief functions as weighted opinions, evidence masses regarding the association of known objects and observations

are expressed in a common frame of discernment (Gruyer and Berge-Cherfaoui, 1999, Gruyer and Berge-cherfaoui, 1999). Afterwards, these masses are combined using fusion operators from the TBM. The association decisions are made following the degree of belief , plausibility or using a pignistic transformation (Mercier and Jolly, 2011). These works represent an alternative to the widely used probabilistic framework. Their observations-to-track interpretation based on TBM has paved the road for future solutions. However, these solutions are hardly implemented in real-time systems due to the hypothesis explosion directly related to the increasingly number of observations in a real driving scenario.

Multiple Hypothesis Tracking (MHT) was originally developed by Reid (1979). This technique has been successfully used by many DATMO solutions to solve data association problems (Burlet *et al.*, 2007, Cox and Hingorani, 1996, Vu, 2009). It stores all the possible hypotheses associations from several frames until the information is considered enough to take an association decision (Blackman, 2004). Although its results are promising, the computational time increases according to the number of observations in the scenario. In cluttered environments this disadvantage can become a main drawback to consider all the new observations. To solve this problem, it is advised to reduce the number of stored hypotheses, integrate a pruning technique and include a gating process (Baig, 2012).

2.3.4 Filtering

In a simplified scenario, we can assume that a moving object motion can be represented by a single motion model, in this case filtering can be performed by using state of the art techniques such as Kalman filter, Extended KF, Unscented KF or Particle filter to estimate object dynamic states (Farmer *et al.*, 2002, Khan *et al.*, 2004). Unfortunately, real-world objects do not behave that simple and can change their dynamic behaviour from one instant to another, e.g., a vehicle stopping, accelerating, or keeping a constant velocity. Therefore, a multiple dynamics model is required to define a moving object motion. Hence, the problem of filtering involves not only estimating dynamic object states but also estimating its corresponding motion modes at each time. While dynamic states are continuous variables, motion modes are discrete variables. These filtering techniques are sometimes called *switching motion models* techniques (Wang *et al.*, 2007).

Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements z observed over time and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. Usually, measurements are noisy or inaccurate. Kalman filter operates

recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. Kalman filtering has been applied several times as the basis for intelligent driving solutions where the unknown variables represent the state of the moving objects. The Kalman filter only deals with linear systems. Regarding these limitations, many variants have been developed: Extended Kalman Filter, to deal with non-linear systems; and for highly non linear systems, Unscented Kalman filter (Beymer *et al.*, 2003, Cuevas *et al.*, 2005, Welch and Bishop, 1995).

Particle filters estimate the posterior density of the state-space by directly implementing the Bayesian recursion equations. These methods use a set of particles to represent the posterior density. Compared to basic Kalman filter, Particle filters make no restrictive assumption about the dynamics of the state-space or the density function. They provide a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. The state-space model can be non-linear and the initial state and noise distributions can take any form required (Guo, 2008, Khan *et al.*, 2004, Thrun, 2000).

The tracking problem of a single moving object can be defined in probabilistic terms as:

$$P(x_t, \mu_t | z_{0:t}), \tag{2.3.3}$$

where x_t is the state of the object, μ_t is its motion model at time t and $z_{0:t}$ are the observations of the object state until time t. Here, μ_t is defined a priori.

Tracking of highly mobile objects (i.e., low, high or non-manoeuvring) is considered a non-linear or non-Gaussian problem. A single model filtering technique cannot sufficiently model the complex variables which represent the state of the tracked objects. In the common structure of multiple model approach, it is assumed that the mode of the system obeys one of a finite number of models in which the system has both continuous nodes as well as discrete nodes. Figure 2.3 shows a graphical representation of the multiple motion model object tracking. A popular technique for tracking multiple modal manoeuvring objects is the Interactive Multiple Models (IMM). The IMM approach overcomes the problem of dealing with motion model uncertainty by using more than one motion model. In IMM, the state estimate at time t is computed under each possible current model using n different filters where each filter uses a suitable mixing of the previous model-conditioned estimates as the initial condition. The final object model is obtained by merging the estimations of all the n elemental filters (Dang *et al.*, 2004, Farmer *et al.*, 2002, Kaempchen, 2004).

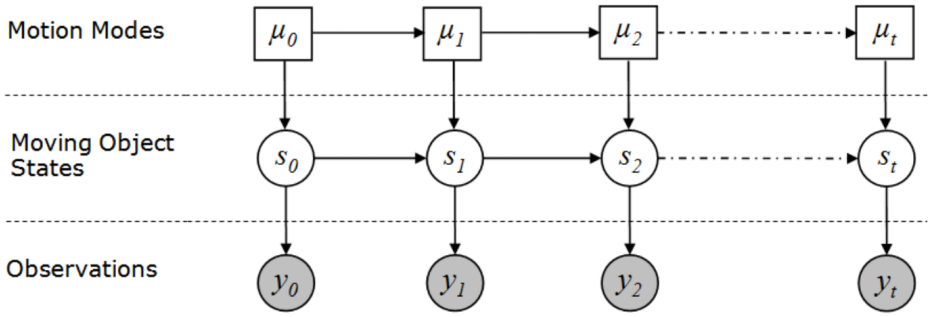


Figure 2.3: Graphical representation of the problem of multiple motion model object tracking (Vu, 2009).

Motion Models

The formulation of moving object tracking presented in Equation 2.3.3 and illustrated in Figure 2.3 shows that the motion model or a set of motion models need to be selected a priori in order for the tracking process to take it into account. Wang *et al.* (2007) have shown how the performance of moving object tracking is related with the selection or inclusion of the selected motion models. Figure 2.4 illustrates the different results obtained by using a single model tracking versus a multiple model tracking.

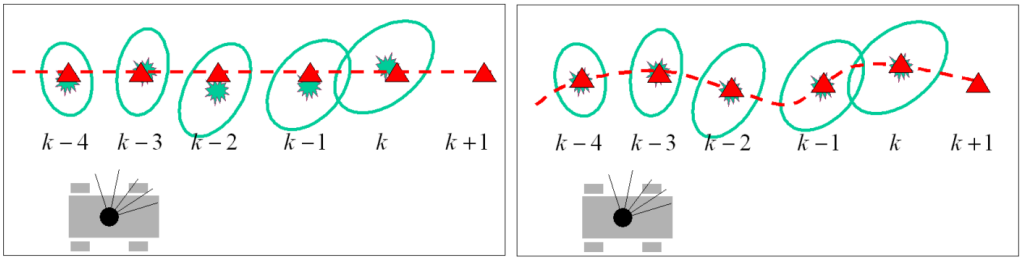


Figure 2.4: Results of moving object tracking based on (left) a single motion model and (right) multiple motion models (Wang *et al.*, 2007).

2.4 Object classification

Moving object classification is not usually mentioned as a fundamental process within the DATMO problem. However, knowing the class of the detected moving objects provides important information for future components of the perception task. Usually, object classification is performed in the late steps of the DATMO component; it is seen as an addition to the final output. We believe that classification provides early knowledge about the detected objects. Also, object classification represents a key factor to improve inner modules of the DATMO task. Moreover, it can improve the general performance

of the whole perception task. Specifically, we will present two fusion approaches at classification and detection levels in Chapter 4 and Chapter 5, where classification information is taken into account as part of the moving object state estimation, representing an important factor within our proposed composite object representation.

Object classification is the task to assign an object observation to one or more classes or categories. These set of possible classes represents the objects of interest that a perception application intends to recognize. In the case of driving applications, the classification process can be: *supervised*, where some external mechanism provides information on the correct classification; *unsupervised*, where the classification must be done entirely without reference to external information; or *semi-supervised*, where parts of the observations are labelled by an external mechanism.

Object classification is a wide field of research. Several improvements have been achieved not only for automotive applications but for a multitude of areas. As in moving object detection, most of the state-of-the-art approaches can be classified according to the input information that is used to classify the object observations. However, they can also be classified according to the specific object class they intend to identify. This means that class information can be extracted, in a minor or major way, from different types of sensor measurements which provide discriminating evidence to classify specific kind of objects. For example, lidar sensor measurements provide a good estimation of the width of the detected object; width estimation gives an idea of the class of detected object. Radar measurements provide an estimation of the relative speed of the detected object (target) which can be used to differentiate potential objects of interest, such as vehicles. Camera images are used to extract appearance features from areas of interest (image patches) and use these features to decide if this area contains an object of interest.

The most common approach to object classification is using active sensors such as lasers, lidar, or millimeter-wave radars. Active sensors can detect the distance from the demonstrator to a detected object by measuring the travel time of a signal emitted by the sensors and reflected by the object. Their main advantage is that they can accurately measure specific quantities (e.g., distance and width) directly requiring limited computing resources. Prototype vehicles employing active sensors have shown promising results. However, active sensors have several drawbacks, such as low spatial resolution, and slow scanning speed. In addition, high-performance laser range scanners are not affordable for commercial applications. Moreover, when a large number of vehicles are moving simultaneously in the same direction, interference among sensors of the same type poses a big problem.

Cameras are usually referred to as passive sensors because they acquire data in a non-intrusive way. One advantage of passive sensors over active sensors is cost. With the introduction of inexpensive cameras, we can have almost a complete surrounding view of the vehicle demonstrator, e.g. forward and rear facing. Camera sensors can be used to more effectively track cars entering a curve or moving from one side of the road to another. Visual information extracted from cameras can be very important in a number of related driving applications, such as lane detection, traffic sign recognition, or object classification (e.g., pedestrians, vehicles), without requiring any modifications to road infrastructures. However, object classification based on camera sensors is very challenging due to huge within-class variabilities. For example, vehicles may vary in shape, size, and color. Vehicles' appearances depend on their pose and are affected by nearby objects. Additionally, illumination changes, complex outdoor environments (e.g., illumination conditions), unpredictable interactions between traffic obstacles, and cluttered background make camera-based classification a more challenging task.

In the next subsections we will speak about the state-of-the-art approaches proposed to classify the two main obstacles that are usually present in a common driving scenario: pedestrians and vehicles.

2.4.1 Vehicle classification

Most of the methods reported in the literature for vehicle classification follow two basic steps: Hypothesis generation, where the locations of possible vehicles in an image or lidar data are hypothesized; and hypothesis verification, where tests are performed to verify the presence of vehicles within generated hypotheses.

Hypothesis generation

The objective of the Hypothesis generation process is to find candidate vehicle locations in sensor data (images, lidar scans) for further verification. Some methods use *a priori* knowledge to hypothesize vehicle locations, e.g. image patches. Other methods use motion assumptions to detect vehicles, e.g. moving clusters of lidar beams, optical flow from image sequence.

Knowledge about vehicle appearance may include: symmetry, geometrical features (i.e., corners, edges, width), texture, geometrical relations (e.g., lights position, tire number).

Images of vehicles observed from rear or frontal perspectives can be considered (most of the times) symmetrical in the horizontal plane. Symmetry has been used as a

main feature to detect vehicles in images. Bertozzi *et al.* (2000) provide an interesting review of most of the common methods used to extract symmetry from vehicle images. Some methods search for leading vehicles in the same lane by means of a contour following algorithm which extracts the left and right object boundary and then verifies vertical axis symmetry. Alternative solutions focus on the shadow area beneath the frontal vehicles. This area is segmented with a contour analysis and then the method verifies two lateral boundaries for vehicles in its own lane and one for those in the neighboring lanes. Chavez-Garcia *et al.* (2012) use radar output to identify moving vehicles, this output is then used to localize patches inside the camera image to confirm vehicle detection based on horizontal edges usually present in the rear and frontal bumpers of a vehicle.

Color cameras are not very common in outdoor vehicle applications, but some works have proposed interesting solutions for vehicle classification based on color signatures. Buluswar and Draper (1998) present a technique based on multivariate decision trees, this technique linearly approximates non-parametric functions to learn the color of specific target objects (cars) from training samples, afterwards this technique classifies the pixels from observations based on the approximated functions. Dickmanns *et al.* (1994) propose a vehicle classifier based on the idea that the area underneath a vehicle is distinctly darker than any other areas on an asphalt paved road. The intensity of the shadow depends on the illumination of the image, which depends on weather conditions. Therefore, methods focused on set or learned threshold values for shadow areas were proposed (Tzomakas and Seelen, 1998). Following the idea that vehicles' shapes are limited by corners, Vu (2009) proposes a classification method to preliminarily identify vehicles based on the visible shape of moving objects represented by lidar clusters. Clusters of vehicles were assumed to be composed of visible corners. Bertozzi *et al.* (1997) propose a method to find matching corners in camera images. To do so, the method uses a fixed set of corner templates usually found in a vehicle image. Srinivasa (2002) extracts vertical and horizontal edges using the Sobel operator. Then, two edge-based constraint filters are applied on those edges to segment vehicles from background.

Shadow, color, texture and features detection depends highly on weather conditions. Distinctive parts of the car, such as lights or plate reflection are good cues to identify vehicles. Cucchiara and Piccardi (1999) present a method based on morphological analysis to detect vehicle light pairs and therefore infer the class of the moving object detection with this morphological composition. In order to do so, this method estimates the size of the object to provide the vehicle hypotheses.

Visual object recognition is considered a difficult problem and therefore great efforts are expended to discover visual cues to represent objects of interest (such as vehicles). Invariance of the visual object representation and preservation of image variation (e.g. variation in position, scale, pose, illumination) are fundamental problems in a visual object recognition system. Visual descriptors describe elementary characteristics of an image, such as the shape, the color, the texture or the motion, among others. Literature about visual feature extraction and description is overwhelming. We decided to mention the most common used features (Pinto *et al.*, 2011).

- Scale-invariant feature transform (SIFT) is an algorithm in computer vision to detect and describe local features in images. SIFT's descriptor is able to find distinctive key-points that are invariant to location, scale, rotation, and robust to affine transformations (changes in scale, rotation, shear, and position) and changes in illumination, they are usable for object recognition.
- Pyramid Histogram Of visual Words (PHOW) is a spatial pyramid representation of appearance. To compute these features, a dictionary of visual words is generated by quantizing the set of descriptors (e.g., SIFT descriptors) with a clustering method.
- Pyramid Histogram Of Gradients (PHOG) is a spatial pyramid representation of shape based on Histogram of Oriented Gradients (HOG) of edges extracted with an edge operator such as Canny, Sobel, among others. An angular range and a number of quantization bins are fixed to build the descriptor.

Visual descriptors are just the signatures of the objects of interest. Machine Learning methods for information classification are used to train a classifier using the visual descriptors from training examples. Works like the ones proposed in (Chavez-Garcia *et al.*, 2012) and (Chavez-Garcia *et al.*, 2013) train a vehicle classifier using descriptors based on edge key-points. Here, the hypotheses are generated using regions of interest provided by lidar moving object detection and then represented using an adaptation of the HOG descriptor.

Vehicle solutions employing symmetry, colors, or corners information to generate hypotheses are most effective in simple environments with distinctive vehicle features and without clutter. Employing these cues in complex environments, such as daily urban scenarios with several moving and static obstacles would introduce many false hypotheses. Colour information has not been used in outdoor environments due to a high correlation between the object color and the illumination, reflectance and visibility of the vehicle. Feature descriptors are by far the more used approach to describe a

vehicle. However, there is no general visual feature nor descriptor to perfectly discriminate a vehicle from non-vehicle obstacles. In this dissertation we use visual descriptors to represent cars and trucks from different points of views. Nevertheless, we also use shape features to extract class information from lidar data. Further detail about our proposed implementation is given in Chapter 3.

Hypothesis verification

Hypothesis verification consists of testing the generated hypotheses to verify their correctness, i.e. that the generated hypotheses represent an object of interest. According to the method used to represent the generated hypothesis, the verification step can be template-based or appearance-based. Template-based methods use predefined patterns from the vehicle class and perform correlation. Appearance-based methods learn the characteristics of the vehicle class from a set of training images which should capture the variability in vehicle appearance, e.g. machine learning classifiers. Classifiers learn the decision boundary between one class and the others, e.g. car, truck, non-vehicle.

Template-based methods use predefined patterns of the vehicle class and perform correlation between the image and the template. For example, the presence of the rear wind-shield, plates or night lights in the generated hypotheses (Cucchiara and Piccardi, 1999, Parodi and Piccioli, 1995).

Using appearance methods, most of the vehicle recognition systems assume the two-class restriction for hypothesis verification: vehicle, non-vehicle. Appearance-based methods learn the characteristics of vehicle appearance from a set of positive training images which capture the variability in the vehicle class (Sun *et al.*, 2004). Moreover, the variability of the non-vehicle class is also modelled using a negative training set of images. The decision boundary between the vehicle and non-vehicle class is learned either by training a classifier or by modelling the probability distribution of the features in each class. Training techniques literature is as big as feature selection literature. Depending on technique used, training could be computationally expensive, therefore many modern solutions train the object classifiers off-line. However, hypothesis generation and verification is performed on-line.

Appearance-based methods are becoming more popular due to the exponential growth in processor speed. Analysing the right combination of visual descriptor and training algorithm is usually done by using different training data sets and strongly depends on the scenario application. In Chapter 3, we describe the algorithm of our

vehicle classifier and present an analysis of the descriptor/classifier combination we choose for our moving object perception application. Later on, in Chapters 4 and 5, we use the trained classifier and visual descriptor in our two on-line fusion approaches.

2.4.2 Pedestrian classification

Detecting the presence of both stationary and moving people in a specific area of interest around the moving host vehicle represents an important objective of an intelligent vehicle system, also known as pedestrian protection systems. As in vehicle detection, pedestrian detection involves many challenges. The appearance of pedestrians exhibits very high variability since they can change pose, wear different clothes, carry different objects, and have a considerable range of sizes according with the view perspective. Driving scenarios such as cluttered urban areas present a wide range of illumination, weather conditions, and occlusions by moving or static obstacles which varies the quality of the sensed information. Most of the driving scenarios are highly dynamic, since both the vehicle demonstrator and the moving pedestrians are moving and interacting. The detection of pedestrians is a task which requires high performance and accuracy in terms of reaction time and robustness.

Contrary to surveillance applications, pedestrian detection for driving systems can not use the simplification of a static camera which allows the use of common background subtraction techniques.

Pedestrian detection for driving applications has become a widely researched topic due to its direct application in on-board safety systems to avoid collisions or mitigate unavoidable damage to pedestrians or the driver of the intelligent vehicle. We follow the same structure as in Section 2.4.1 to mention the state-of-the-art methods employed to detect pedestrians in driving scenarios: firstly, we review methods that generate hypotheses of pedestrians; secondly, we review the hypotheses verification approaches.

Hypothesis generation

Some works propose pedestrian detector solutions using measurements from active sensors, such as lidar and radar. Chavez-Garcia *et al.* (2012) hypothesize pedestrians in lidar data, assuming that such moving objects are present where small moving groups of lidar hits appear. Vu (2009) and Chavez-Garcia *et al.* (2013) use radar measurements to confirm the pedestrian hypotheses by finding a correlation between radar targets and small lidar hit groups. However, most of the research on pedestrian detection is made by using passive sensors (e.g., camera and infra-red camera).

Existing vision-based approaches for pedestrian classification can be divided into two categories: monocular-based and stereo-based. In monocular-based approaches, the algorithms are mainly based on the study of visual features to represent pedestrians. A comparative study of different monocular camera's pose estimation approaches is presented in (Viola and Jones, 2001a). It includes horizontal edges, features-based, and frame difference algorithms. Mateo Lozano and Otsuka (2008) presented a probabilistic framework for 3D geometry estimation based on a monocular system. Here, a training process, based on a set of manually labelled images, is applied to form an estimation of the horizon position and camera height. Regarding stereo-based pose estimation, Labayrade *et al.* (2007) introduced v-disparity space, which consists of accumulating stereo disparity along the image y-axis in order to compute the slope of the road and to point out the existence of vertical objects when the accumulated disparity of an image row is very different from its neighbors. Andriluka *et al.* (2008) proposed fitting 3D road data points to a plane, whereas Singh *et al.* (2008) used an Euler spiral. In euclidean space, classical test squares fitting approaches can be followed, while in the v-disparity space, voting schemes are generally preferred. Leibe *et al.* (2004) proposed the use of pedestrian location hypotheses together with depth cues to estimate the ground plane, which is used to reinforce new detections' tracks. This approach is known as *cognitive feedback*, because a loop is established between the classification and tracking modules, and the ground plane estimation.

Stereo-based approaches provide more robust results in camera pose estimation than monocular approaches. Horizon-like 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). 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, Euler spiral, and any free form surface approximation. However, monocular-based approaches represent the current state-of-the-art configuration in vehicle demonstrators due to affordability and almost straightforward configuration compared with the calibration process of a stereo-vision set-up.

Foreground Segmentation is considered a candidate generation process. It extracts regions of interest (ROI) from the image to be sent to the classification module, avoiding as many background regions as possible. The importance of this process is to avoid missing pedestrians; otherwise, the subsequent modules will not be able to correct the

error. Some works used fixed based aspect ratio, size and position to generate the ROI to be considered to contain a pedestrian (Baig *et al.*, 2011, Marchal *et al.*, 2005). Recently, works presented in (Chavez-Garcia *et al.*, 2012, 2013) have proposed the generation of ROI as a lidar-based procedure. Here, clusters of moving lidar points with a fixed size are transformed into bounding boxes on the camera image to represent hypotheses of pedestrians.

The first idea of candidate generation is an exhaustive scanning (sliding window) approach that selects all of the possible candidates in an image according to fixed sizes, without explicit segmentation (Chan and Bu, 2005). Usually, the fixed size of a pedestrian is modified during ROI generation using size factors. This idea has two main drawbacks: the number of candidates can be very large, which makes it difficult to fulfil real-time requirements; and many irrelevant regions are passed to the next module, which increases the potential number of false positives. As a result, other approaches are used to perform explicit segmentation based on camera image, road restriction, or complementary sensor measurements.

The most robust techniques to generate ROIs using camera data are biologically inspired. Milch and Behrens (2001) select ROIs according to color, intensity and gradient orientation of pixels. Dollár *et al.* (2012) review in detail current state-of-the-art intensity-based hypothesis generation for pedestrian detection. Some of these methods involve the use of learning techniques to discover threshold values for intensity segmentation. Optical flow has been used for foreground segmentation, specially in the general context of moving obstacle detection. Franke, U. and Heinrich (2002) propose to merge stereo processing, which extracts depth information without time correlation, and motion analysis, which is able to detect small gray value changes in order to permit early detection of moving objects, e.g. pedestrians.

Hypothesis verification

The pedestrian classification module, or pedestrian hypotheses verification, receives a list of ROIs that are likely to contain a pedestrian. In this stage, ROIs are classified as pedestrian or non-pedestrian. Its goal is minimizing the number of false positives and mis-classifications. The simplest approach is the binary shape model in which an upper body shape is matched to an edge modulus image by correlation after symmetry-based segmentation (Broggi *et al.*, 2000). This method is considered as the silhouette matching approach.

Appearance based methods define a space of image features (also known as de-

scriptors), and a classifier is trained by using ROIs known to contain examples (pedestrians) and counterexamples (non-pedestrians). Gavrila *et al.* (2004) propose a classifier that uses image gray-scale pixels as features and a neural network with local receptive fields as the learning machine that classifies the candidate ROIs. Zhao and Thorpe (1999) use image gradient magnitude and a feedforward neural network to classify images patches (ROIs). Papageorgiou and Poggio (2000) introduce Haar wavelets (HWs) as features to train a quadratic support vector machines (SVMs) with front and rear viewed pedestrians samples. HWs compute the pixel difference between two rectangular areas in different configurations, which can be seen as a derivative at a large scale. Viola and Jones (2001b) propose AdaBoost cascades (layers of threshold-rule weak classifiers) as a learning algorithm to exploit Haar-like features for surveillance-oriented pedestrian detection. Dalal and Triggs (2005) present a human classification scheme that uses SIFT-inspired features, called histograms of oriented gradients (HOG), and a linear SVM as a learning method. An HOG feature divides the region into k orientation bins, also defines four different cells that divide the rectangular feature, and then a Gaussian mask is applied to the magnitude values in order to weight the center pixels, and the pixels are interpolated with respect to pixel location within a block. The resulting feature is a vector of dimension 36 containing the summed magnitude of each pixel cells, divided into 9 ($kvalue$) bins. These features have been extensively exploited in the literature (Dollár *et al.*, 2012). Recently, Qiang *et al.* (2006) and Chavez-Garcia *et al.* (2013) use HOG as a weak rule for AdaBoost classifier, achieving the same detection performance, but with less computation time. Maji *et al.* (2008) and Wu and Nevatia (2007) proposed a feature based on segments of lines or curves, and compared it with HOG using AdaBoost and SVM learning algorithms.

Methods that exploit the appearance of pedestrians indicate a promising direction of research. This idea is reinforced by the outstanding development of new learning algorithms and visual features to use as input for these algorithms. Moreover, the increasing computational power allows to overcome early computing time restrictions from visual features calculation. We follow this path not only for pedestrian classification but also for vehicle classification. The number of current visual descriptors in the literature is quite large. Besides, from the proposed works mention above, we notice that the feature and descriptor selection depends on the objects of interest. We decided to use HOG-based descriptors to represent our objects of interest by extracting edge-based features. HOG descriptors have shown promising results for car and pedestrian detection (Dollár *et al.*, 2012). Further detail about the vision-based object classifiers is given in Chapter 3.

2.5 Multi-sensor fusion

The data fusion process model was proposed by the Joint Directors of Laboratories Data Fusion Working Group. This model represents a generic layout for a data fusion system and was designed to establish a common language and model as a basis over many data fusion techniques have been created (Hall and Llinas, 1997). The fusion model defines relationships between the data sources and the processes carried out to extract information. According to Hall and Llinas (1997), between the data extraction and the final information given to decision stages different processing levels can be identified:

- Source processing: creates preliminary information from raw data.
- Object refinement: refines the preliminary information to identify objects.
- Situation refinement: establishes the relations among identified objects.
- Threat refinement: tries to infer details about future states of the system.
- Process refinement: analyses the performance of the previous levels and determines if they can be optimized.
- Data management: takes care of the storage management of the processed data.

Multi-sensor data fusion is the process of combining several observations from different sensor inputs to provide a more complete, robust and precise representation of the environment of interest. The fused representation is expected to be better than the one provided by the individual inputs.

In the following sections, we will review the state-of-the-art works on multi-sensor fusion from two perspectives. First, we will analyse the most common methods used to fuse sensor information highlighting their advantages and drawbacks. Second, we will consider the different levels where fusion can be performed inside the perception problem. The last perspective allows us to analyse the advantages of performing fusion at each level and also to focus on the unsolved issues of the related fusion approaches.

2.5.1 Fusion methods

The most common approaches for multi-sensor fusion are based on probabilistic methods, however methods based on the theory of evidence proposed an alternative not

only to multi-sensor fusion but to many modules of vehicle perception. In the following subsections we will review the main fusion approaches within these two alternatives. Afterwards, we will describe the different architectures of multi-sensor fusion for intelligent vehicle perception.

Probabilistic methods

At the core of the probabilistic methods lies Bayes' rule, which provides means to make inferences about an event or object described by a state x, given a measurement observation z. The relationship between x and z is encoded in the joint probability distribution $P(x, z)$ which can be expand as follows:

$$P(x, z) = P(x|z)P(z) = P(z|x)P(x), \tag{2.5.1}$$

therefore, we obtain Bayes' rule in terms of the conditional probability $P(x|z)$:

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)}, \tag{2.5.2}$$

where $P(x)$ represents the prior probability and encodes prior information about the expected values of x. To obtain more information about the state x, an observation z is made. These observations are modelled in the form of a conditional probability $P(z|x)$ which describes, for each fixed state x, the probability that the observation z might be made. New likelihoods associated with the state x are computed from the product of the original prior information ($P(x)$) and the information gained by an observation ($P(z|x)$). This information is encoded in the posterior probability $P(x|z)$ which describes the likelihoods associated with x given the observation z. In this fusion process, the marginal probability $P(z)$ is used to normalize the posterior probability. $P(z)$ plays an important role in model validation or data association as it provides a measure of how well the observation is predicted by the prior. Bayes' rule provides a principled means of combining observed information with prior beliefs about the state of the world.

In driving applications, conditional probability $P(z|x)$ plays the role of a sensor model. Whilst building a sensor model, the probability $P(z|x)$ is constructed by setting the value of $x = x'$ and then calculating $P(z|x = x')$. Alternatively, when this sensor model is used and observations are made, $z = z'$ is fixed and a likelihood function $P(z = z'|x)$ is inferred. The likelihood function models the relative likelihood that different values of x produce the observed value of z. The product of this likelihood with the prior, both defined on x, gives the posterior or observation update $P(x|z)$.

Using Bayes' rule for multi-sensor fusion requires conditional independence and is represented as follows:

$$P(x|Z^n) = CP(x)\prod_{i=1}^{n} P(z_i|x), \tag{2.5.3}$$

where C is a normalizing constant. Equation 2.5.3 states that the posterior probability on x given all observations Z^n, is proportional to the product of prior probability and individual likelihoods from each information source. Therefore, recursive form of Bayes' rules is defined as:

$$P(x|Z^k) = \frac{P(z_k|x)P(x|Z^{k-1})}{P(z_k|Z^{k-1})}, \tag{2.5.4}$$

where $P(x|Z^{k-1})$ stores all past information. Hence, when the next piece of information $\mathrm{P}(z_k|x)$ arrives, the previous posterior takes on the role of the current prior and, after normalizing, the product becomes the new posterior.

Occupancy grids

Probabilistic occupancy grids (POGs) are conceptually the simplest approach to implement Bayesian data fusion methods. Although simple, POGs can be applied to different problems within the perception task: e.g., mapping (Vu, 2009, Wang *et al.*, 2007), moving object detection (Baig *et al.*, 2011), sensor fusion (Grabe *et al.*, 2012).

In mapping, the environment of interest is discretized into a grid of equal sized spatial cells. Each cell carries the probability value that represents the occupancy state of the cell $P(x_{i,j})$, usually a cell can be *empty* or *occupied*. The goal is to maintain a probability distribution on possible state values $P(x_{i,j})$ at each grid cell. Once the state has been defined, Bayesian methods require sensor models or likelihood functions to populate the grid. This requires the definition of the probability distribution $P(z|x_{i,j})$ mapping each possible grid state $x_{i,j}$ to a distribution of observations. Usually, this is implemented as an observation grid per sensor input, so that for a specific observation $z = z'$, a grid of likelihoods over the occupancy state of each $x_{i,j}$ is produced.

One can notice that the computational cost of maintaining an updated a fused occupancy grid is strongly related with the resolution and size of the environment. Grid based fusion is appropriate to situations where the domain size and dimension are modest or when assumptions about the environment can be made to reduce the size of the grid or updating cost. In such cases, grid based methods provide straightforward and effective fusion algorithms. Grid based methods can be improved in a number of ways: hierarchical (quadtree) grids (Bosch *et al.*, 2007), irregular (triangular, pentagonal) grids (Azim and Aycard, 2012) or working on local map assumptions to avoid a

global map updating process (Vu, 2009).

Kalman filter

The Kalman filter (KF) is a recursive linear estimator which successively calculates an estimate for a continuous valued state x, that evolves over time, on the basis of periodic observations Z of the state. KF employs an explicit statistical model of how the parameter of interest $x(t)$ evolves over time and an explicit statistical model of how the observations $z(t)$ are related to this parameter. The gains employed in a KF are chosen to ensure that the resulting estimate $\hat{x}(t)$ minimizes the mean-squared error, hence representing the conditional mean $\hat{x}(t) = E[x(t)|Z^t]$. KF features make it suited to deal with multi-sensor estimation and data fusion problems. First, its explicit description of processes and observations allows a wide variety of different sensor models to be incorporated within the basic algorithm. Second, the consistent use of statistical measures of uncertainty makes it possible to quantitatively evaluate the role each sensor plays in overall system performance. In addition, the linear recursive nature of the algorithm ensures that its application is simple and efficient (Baig *et al.*, 2011, Wang *et al.*, 2007).

A basic assumption in the derivation of the Kalman filter is that the random variables describing process and observation noise are all Gaussian, temporally uncorrelated and zero-mean. If these constraints are not fulfilled results produced by the Kalman filter might be misleading. In these cases, more sophisticated Bayesian filter are usually applied. The extended Kalman filter (EKF) is an extension of the Kalman filter that can be employed when at least the state model or the observation model are nonlinear (Kalman, 1960). EKF is considered a non-linear version of KF. Another non-linear version of KF is the unscented Kalman filter (UKF). In the UKF, the probability density is approximated by a deterministic sampling of points which represent the underlying distribution as a Gaussian. The non-linear transformation of these points is intended to be an estimation of the posterior distribution. The transformation is known as the unscented transform. In practice, UKF tends to be more robust and more accurate than the EKF in its estimation of error (Julier and Uhlmann, 2004).

Monte Carlo methods

Monte Carlo (MC) methods describe probability distributions as a set of weighted samples of an underlying state space. MC filtering then uses these samples to simulate probabilistic inference usually through Bayes' rule. Many samples or simulations are performed. By studying the statistics of these samples as they progress through the inference process, a probabilistic behavior of the process being simulated is discovered (Zhu *et al.*, 2000).

MC methods are well suited for problems where state transition models and observation models are highly non-linear. The reason for this is that sample-based methods can represent very general probability densities. In particular, multi-modal or multiple hypothesis density functions are well handled by Monte Carlo techniques (Vu, 2009). MC methods are considered to span the gap between parametric and grid-based data fusion methods. However, MC methods might be inappropriate in problems where the state space is of high dimension. The reason for this is that the number of samples required to obtain an accurate model increases exponentially with state space dimension. Fortunately, the dimensionality growth can be limited by marginalizing out states that can be modeled without sampling.

Limitations of probabilistic methods

Uncertainty representation is an important factor for the problem of information representation and therefore information fusion. Probabilistic methods are suited for random uncertainty representation, but they do not propose an explicit representation of imprecision. Regarding their perception limitations, we can list the main issues of probabilistic methods for information fusion.

1. Complexity: the need to specify a large number of probabilities to be able to apply probabilistic reasoning methods correctly.
2. Inconsistency: the difficulties involved in specifying a consistent set of beliefs in terms of probability and using these to obtain consistent deductions about the events or states of interest.
3. Precision of models: the need to have precise specifications of probabilities about barely known events.
4. Uncertainty: the difficulty in assigning probability in the face of uncertainty, or ignorance about the source of information.

We can mention three main theories that intend to overcome the previous limitations of probabilistic methods: interval calculus, fuzzy logic and theory of evidence.

Interval calculus

Interval calculus (IC) is based on the idea of representing uncertainty using an interval to bound true parameter values. Compared to probabilistic methods, using IC has potential advantages. Intervals provide a good measure of uncertainty in situations where there is a lack of probabilistic information, but in which sensor and parameter

error is known to be bounded. In IC techniques, the uncertainty in a parameter x is simply described by a statement that the true value of the state x is known to be bounded from below by a, and from above by b, where $x \in [a, b]$. It is important that no other additional probabilistic structure is implied, in particular the statement $x \in [a, b]$ does not necessarily imply that x is equally probable (uniformly distributed) over the interval $[a, b]$.

IC methods are sometimes used for object detection. However, they are not generally used in data fusion due to: the difficulty to get results that converge to a desired value; and the difficulty to encode dependencies between variables which are at the core of many data fusion problems, e.g. variables defining the state and appearance of a moving object.

Fuzzy Logic

Fuzzy Logic has been known as a popular method for representing uncertainty in control and data fusion applications. It deals with reasoning that is approximate rather than exact. Fuzzy reasoning is based on degrees of truth instead of absolute values (Zimmermann, 2010). Degrees of truth cannot be associated with probabilities, they are conceptually distinct: fuzzy truth represents membership in vaguely defined sets, not the likelihood of some event or condition. Fuzzy logic provides a strict mathematical framework in which vague conceptual events can be precisely and rigorously represented and studied. It can also be considered as a modelling language, well suited for situations in which fuzzy relations, criteria, and events exist. Whilst in a classic logic set, membership is binary, i.e., a variable is either in the set or not in the set; in fuzzy sets, membership is based on a degree between the possible absolute values.

Logic as a base for reasoning can be distinguished essentially by three topic-neutral items: truth values, vocabulary (operators), and reasoning procedures (tautologies, syllogisms). In dual logic, truth values can be *true* or *false* and operators are defined by boolean truth tables. In fuzzy logic, the truth values are no longer restricted to the two values *true* and *false* but are expressed by linguistic variables *true* and *false*.

Although fuzzy logic is widely used in control applications, in sensor fusion for driving applications its limitations are considered relevant. It has limited ability to learn membership functions. Besides, determining good membership functions and fuzzy rules are not straightforward, becoming more complex as the number sensor inputs or objects of interest increase. Additionally, verification and validation of the fuzzy system requires extensive testing which is especially important in systems where

safety plays an important factor (Subramanian *et al.*, 2009).

Evidence Theory

Evidence Theory (ET), also known as the Dempster-Shafer theory of evidence, has been used as an important alternative to probabilistic theory (Dempster, 2008). In particular, it has shown important success in automated reasoning applications, such as intelligent driving systems. Evidential reasoning is qualitatively different from both probabilistic methods and fuzzy set theory. Let us consider a universal set Ω of all possible hypotheses of an event x. In probability theory or fuzzy set theory, a belief mass (likelihood of a hypothesis) may be placed on any element $a \in \Omega$ and indeed on any subset $A \subseteq \Omega$. In evidential reasoning, belief mass can not only be placed on elements and sets, but also sets of sets. Specifically, while the domain of probabilistic methods is all possible subsets Ω, the domain of evidential reasoning is the power set 2^{Ω}. Briefly, Dempster-Shafer's model aims at quantifying degrees of belief.

In the field of intelligent vehicle perception there is a variety of *imperfect* information: uncertain or imprecise. For example, object are missing (occlusions), sensor cannot measure all relevant attributes of the object (hardware limitations), and when an observation is ambiguous (partial object detection). Imprecision as well as uncertainty induce some beliefs, some subjective opinions held by an agent at a given time about the real value of a variable of interest (Smets, 1999).

Transferable Belief Model

Transferable Belief Model (TBM) is an interpretation of Dempster-Shafer's theory based on the work of Shafer (1976). Shafer stated that this model is a *purified* form of Dempster-Shafer's model in which any connection with probability concepts has been deleted.

Let the frame of discernment (or hypotheses space) Ω be the set of all possible solutions of a problem, where all of its elements are mutually exclusive. The knowledge of the environment held by the agent Y can be quantified by a belief function with the power set 2^{Ω} as the domain, and the range $[0, 1]$ as image. The TBM is able to explicitly represent uncertainty on a hypothesis from Ω. It takes into account what remains unknown and represents the belief for the hypotheses that are already known.

One of the elements of Ω, denoted ω, corresponds to the actual problem's solution but due to the agent's imprecision, it is not certain about which element. The agent can only express his subjective opinion about the fact that certain subset of Ω might contain ω. The belief $bel(A)$ given by Y at time t to a subset A of Ω expresses the strength of

the agent's belief that ω is an element of A based on the available information by Y at current time t. The degree of belief is usually quantified by a likelihood measure. The Transferable Belief Model concerns the same problem as the one considered by the Bayesian model except it does not rely on probabilistic quantification but on belief functions (Smets, 1990).

TBM is a two-level-based model: in the *credal* level beliefs are quantified by belief functions, where the agent expresses the strength of his beliefs about the fact that ω belongs to some subsets of Ω. It is in this level where the knowledge is stored, updated and combined. The *pignistic* level appears when a decision must be made, the beliefs held at the credal level will transmit the needed information to the pignistic level to make the optimal decisions. This level has no activity when no decision must be made (Smets, 1999). TBM assumes that the beliefs held at credal level are quantified by belief functions (Shafer, 1976).

The TBM postulates that the impact of a piece of evidence on an agent is translated by an allocation of parts of an initial unitary amount of belief among the subsets of Ω. For $A \subseteq \Omega$, $m(A)$ is a part of the agent's belief that supports A (Smets, 1999). The $m(A)$ values, $A \in \Omega$, are called the basic belief masses (*bbm*) and the m function (or *mass function*) is called the *basic belief assignment* (BBA) and is defined as:

$$\begin{aligned} m(A) &\rightarrow [0,1] \\ &\textit{with}: \\ \sum_{A \subseteq \Omega} m(A) &= 1, \\ m(\emptyset) &= 0. \end{aligned} \tag{2.5.5}$$

Every $A \in \Omega$, such that $m(A) > 0$, is called a *focal proposition (or focal element)*. The difference with probability models is that masses can be given to any subsets of Ω instead of only to the elements of Ω as it would be the case in probability theory.

From equation 2.5.5, one can notice that a BBA may support a set A without supporting any of its subsets. This can be seen as a partial knowledge capability. A can be any subset in 2^{Ω}, but there are some cases when the mass function $m(A)$ has a special meaning:

- If there is only one focal set, $m(A) = 1$ for some $A \subseteq \Omega$, then $m(A)$ becomes a categorical mass function. And if $A = \Omega$, *the vacuous function*, this represents the total ignorance.

- If all focal sets are singletons, $m(A) > 0 \rightarrow |A| = 1$, $m(A)$ could be considered a Bayesian mass function.

Let us imagine that there is new information about $\omega \in B \subseteq \Omega$. Hence, this results in an evidence transfer for each $A \subseteq \Omega$ of the *bbm* $m(A)$ initially allocated in A, to $A \cap B \subseteq \Omega$. This is the reason why this model is known as the *Transferable Belief Model* (Smets, 1999).

Evidence reliability is an important concept in the TBM. When the evidence masses assigned in the BBA come from a non-reliable source, these masses should be considered as imprecise. A discounting factor allows to introduce this lack of reliability by weighting the mass assignations.

The original transfer of belief described in the TBM corresponds to the *unnormalized rule of conditioning*, also named Dempster's rule of conditioning. Let us assume that conditioning evidence tells the agent Y that $B \subseteq \Omega$ is true. Hence, we can transfer the original BBA m into an updated BBA m_B as follows:

$$\begin{aligned} m_B(A) &= \sum_{X \subseteq \bar{B}} m(A \cup X),\ A \subseteq B, \\ m_B(A) &= 0,\ A \nsubseteq B, \end{aligned} \tag{2.5.6}$$

where m is a BBA on the frame of discernment Ω.

The *degree of belief*, $bel(A)$, of $A \subseteq \Omega$ quantifies the total amount of justified support given to A. It is obtained by summing all the basic belief masses given to propositions $X \subseteq A$, with $X \neq \emptyset$ (Smets, 1999):

$$\begin{aligned} bel &: 2^{\Omega} \rightarrow [0,1], \\ \text{where } bel(A) &= \sum_{\emptyset \neq X \subseteq A} m(X), \end{aligned} \tag{2.5.7}$$

which is *justified* because $bel(A)$ includes evidence *only* given to specific subsets of A. The function *bel* is called a belief function and satisfies the inequalities proposed by Shafer (1976):

$$\begin{aligned} &A_1, A_2, ..A_n \subseteq \Omega \\ &bel(A_1 \cup A_2 \cup ...A_n) \geq \sum_i bel(A_i) - \sum_{i>j} bel(A_i \cap A_j) ... - (-1)^n bel(A_1 \cap A_2 \cap ...A_n) \\ &\forall n \geq 1 \end{aligned} \tag{2.5.8}$$

Total ignorance is a state of belief that is hard to represent in probability models. This state represents a problem in Bayesian theory when it has to be defined by probability functions (Smets, 1999). In the TBM, total ignorance is represented by a *vacuous* belief function $m(\Omega) = 1$, hence $bel(A) = 0$, $\forall A \subseteq \Omega, A \neq \Omega$, and $bel(\Omega) = 1$. None of the subsets $A \in \Omega$ are supported (except Ω itself) and all subsets receive the same degree of belief, which is the state of total ignorance.

The *degree of plausibility*, $pl(A)$, of $A \subseteq \Omega$ quantifies the maximum amount of potential support that could be given to $A \subseteq \Omega$. It is obtained by adding all the basic belief masses given to propositions X that are compatible with A, this means $X \cap A \neq \emptyset$ (Smets, 1999):

$$\begin{aligned} pl &: 2^{\Omega} \rightarrow [0,1], \\ \text{where } pl(A) &= \sum_{X \cap A \neq \emptyset} m(X) = bel(\Omega) - bel(\bar{A}), \end{aligned} \tag{2.5.9}$$

which means that there are basic belief masses included in $pl(A)$ that could be transferred to non-empty subsets of A if some new information could justify such a transfer. For example if we know that $\bar{A}$ is impossible. The function pl is called a plausibility function.

A *plausibility* function is an alternative representation of the information represented by the belief function over the same BBA. There is a one-to-one correspondence with mass function m, so they never add or loose information:

$$pl(A) = bel(\Omega) - bel(\bar{A}) = \sum_{A \cap B = \emptyset} m(B) \tag{2.5.10}$$

Fusion of evidence

The ability to aggregate multiple evidences from different sources over the same frame of discernment is an important advantage of the evidential theory, which can be interpreted as an information fusion operation. Let us define two evidence distributions m_1 and m_2 with elements from two different sources S_1 and S_2, respectively. These two evidence bodies with [focal] elements $X_1, X_2, ...X_i$ and $Y_1, Y_2, ...Y_j$ can be combined into a new mass function m using a combination operator (combination rule). The most common rule of combination is the one proposed by Dempster (2008). The underlying idea of this rule is that the product of m_1 and m_2 induced by the two bodies of evidence on the same frame of discernment Ω supports $X \cap Y$. Dempster's rule provides a method to compute the orthogonal sum $m = m_1 \oplus m_2$ as follows:

$$
\begin{aligned}
m(A) &= \frac{\sum_{X_i \cap Y_j = A} m_1(X_i) m_2(Y_j)}{1-K} \text{ for } A \subset \Omega, \\
K &= \sum_{X_i \cap Y_j = \emptyset} m_1(X_i) m_2(Y_i), \\
m(\emptyset) &= 0,
\end{aligned}
\tag{2.5.11}
$$

where K is a normalization factor known as the *conflict factor* because it measures the degree of conflict evidence between the bodies of evidence m_1 and m_2. If the value of K is high, the conflict is strong between the sources; therefore, a combination would make no sense.

The normalization factor in Dempster's rule produces convergence toward the dominant opinion between the bodies of evidence to be combined. This means that concordant items of evidence (redundant evidence) reinforce each other by transferring mass in the null set $\emptyset$ to the focal elements. However, when the bodies of evidence are not reliable or the mass functions are imprecise, a conflict mass $m(\emptyset)$ appears. Particularly, when there is a considerable degree of conflict between the sources of evidence, the normalization process inside the Dempster's rule of combination can lead to counter-intuitive results (Smets, 1999). For example, it can lead to assign a high belief to a minority element when no agree is achieved between the evidence sources.

Yager (1985) proposed an alternative to Dempster's rule to avoid counter-intuitive results when a high conflict value is present in the evidence combination. Yager's rule of combination assigns the conflict mass from the null set $\emptyset$ to the ignorance set Ω. Which means that the conflict value is distributed among all the elements of the frame of discernment rather than only the elements with intersections of the combining masses. Yager's rule of combination is defined as follows:

$$
\begin{aligned}
m(A) &= \sum_{X_i \cap Y_j = A} m_1(X_i) m_2(Y_j), \quad A \neq \emptyset, A \neq \Omega, \\
m(\Omega) &= \sum_{X_i \cap Y_j = \Omega} m_1(X_i) m_2(Y_j) + K, \\
K &= \sum_{X_i \cap Y_j = \emptyset} m_1(X_i) m_2(Y_i).
\end{aligned}
\tag{2.5.12}
$$

Analysis of Evidence Theory for Vehicle perception

The first advantage of ET is its ability to represent incomplete evidence, total ignorance and the lack of a need for *a priori probabilities*. Prior to the evidence acquisition process, the total belief is assigned to the ignorance. Therefore, when new evidence is

acquired, this evidence replaces the evidence in the ignorance. Although ET does not require *a priori* knowledge about the environment, implicit knowledge is encoded in the definition of the structure of the frame of discernment. Discounting factors are an important mechanism to integrate the reliability of the sources of evidence into the ET representation, such as sensor performance reliability. Combination rules are very useful tools to integrate information from different bodies of evidence into a more reliable evidence distribution, sensor fusion applications can exploit this ability. The different interpretations of belief functions allow to select the proper decision-making tool for the desired application. Late stages of intelligent vehicle systems, such as reasoning & decision, can integrate evidence distributions into the decision making process using the proper evidence representation (Smets, 2000).

The two main disadvantages of ET rely on the representation and on the combination of evidence. First, when the number of hypotheses is large, ET becomes less computational tractable because the belief functions distribute the belief to the power set of all the hypotheses, 2^{Ω}. However, the application domain may allow to make assumptions to transform Ω into a reduced version of the set of possible hypotheses. This disadvantage might be overcome by the appropriate definition of the frame of discernment. Second, the normalization process derived from the conflict values between evidence bodies leads to counter-intuitive results. Nevertheless, proposals such as the Yager's rule, overcome this limitation by managing the conflict without averaging the combined evidence.

2.5.2 Fusion Architectures

The fusion theories described in previous sections are at the core of many modern multi-sensor fusion techniques. The implementation of these theories is located at different stages inside the multi-sensor fusion systems. We follow the general architecture proposed by Baig (2012) to describe from another perspective the different state-of-the-art methods related to our contributions, and to show graphically where exactly these contributions are taking place inside the DATMO component.

Taking Figure 1.3 as an architectural reference, Figure 2.5 shows the four fusion levels that we consider inside a perception system:

- Low level. Raw data from each sensor is translated into a common representation and then combined to be fused into a representation which is then used to build the map.
- Map level. After solving SLAM for each sensor output, the generated maps are

combined to get a fused map.

- Object detection level. Sensor processes provide lists of moving object in the environment, then fusion is performed between these lists to get an enhanced list objects. Fusion at this level can reduce the number of mis-detections.

- Track level. Lists of moving objects detected and tracked over time by individual sensors are fused to produce the final list of tracks. Fusion at this level can reduce the number of false tracks.

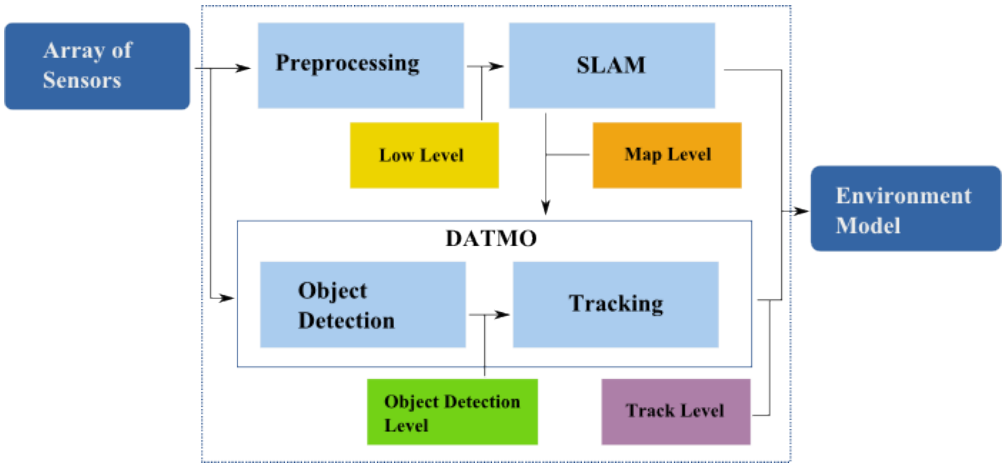


Figure 2.5: Fusion levels within the SLAM and DATMO components interaction.

Low level and map level fusions are performed within the SLAM component, whilst object detection level and track level fusions are performed within the DATMO component.

As stated above, multi-sensor fusion is based on the idea that many sensors can provide complementary information (e.g, lidar and camera sensor) and redundant information (i.e., overlapping field of views). Sensor selection and configuration are processes strongly related with the intelligent vehicle application. Therefore, the variety of sensor set-ups in the proposed multi-sensor fusion methods from the state-of-the-art is vast. In order to narrow the related works we focus on the methods that use lidar, mono-camera and radar sensors. This decision comes from the sensor set-up we use to test our proposed multi-sensor fusion approaches.

As mentioned in Section 1.7, our contributions are placed inside the DATMO component. Therefore, the following section will describe related methods which mainly focus on the last two fusion levels inside DATMO. Further detail about our sensor configuration and contributions is given in Chapters 3, 4 and 5.

Object detection level

An advantage of performing fusion at the object detection level is that the description of the objects, i.e. their properties, can be enhanced by adding knowledge from different sensor sources. For example, lidar data can give a good estimation of the distance to the object and its visible size. Classification information, usually obtained from camera images, may allow to make assumptions about the nature of the detected object to perform model-based tracking and get smooth tracks. An early improvement enrichment of objects' description could allow the reduction of the number of false detections and to involve classification as a key element rather than only an add-on to the perception output.

Baltzakis *et al.* (2003) have developed a fusion technique between lidar and camera data. Using lidar data, they extract line features where each extracted line is supposed to be a wall. Then, they construct a 3D model of walls from all the extracted lines. Each wall is identified by a fixed color. For each pixel in the stereo camera images, they calculate its depth information; then, the pixels are projected on the constructed 3D model to perform a color comparison. If colors match then it is inferred that both camera and laser are seeing the same point otherwise camera is seeing a hanging obstacle in the environment. The computation cost of this technique makes it infeasible for real time applications. Moreover, this method is not applicable to outdoor environments where there is a vast variety of shapes and colors.

Skutek *et al.* (2005) presented a PreCrash application based on multi-sensor data fusion. Two main sensors were used: a laser scanner and two short range radars mounted in the front of the vehicle demonstrator. Their work gives an overview about the whole system structure and the requirements of their PreCrash application. This overview, empirically shows how the system application as the vehicle environment directly influences the choice of sensors. The fusion approach uses the redundancy given by similar data of several sensors based on different physical basics to increase the certainty of object identity and crash-relevant object information in unexpected weather conditions. Two sensor data fusion architectures were proposed. One is directly based on a Kalman Filter. The other is based on a segmented occupancy grid, which is more adapted to the combination of the different data. The lack of appearance information decreases the accuracy of objects' detection. Besides, no moving object classification is performed.

Perrollaz *et al.* (2006) and Labayrade *et al.* (2007) have presented a similar fusion technique between laser and stereo vision for obstacles detection. This technique is

based on stereo vision segmentation and lidar data clustering to perform detection.

Fayad and Cherfaoui (2008) have proposed multi-sensor fusion technique based on the transferable belief model (TBM) framework. This technique is a mixture of object detection and tracking level fusion. This work focuses only in the detection of pedestrians using multiple sensors by maintaining a score for the pedestrian detections. Although the results are promising, this work only considers class information to perform object fusion leaving out location information. Moreover, the extension to detect multiple moving objects classes is not straight forward.

Baig *et al.* (2011) proposed a fusion approach to combine stereo-vision and lidar object detectors. This fusion technique is based on probabilist methods to perform object associations and combine the available information. However, the only source of classification comes from a stereo-vision sensor leaving aside lidar measurements. In this particular implementation, stereo-vision detection has a high rate of mis-detections due to noisy measurements. Reliability from the detection sources was not included. Nevertheless, in order to overcome this issue, a road border detection was proposed to reduce the number of false detections.

Tracking level

Multi-sensor fusion at tracking level requires as input the results from DATMO solutions from each sensor input. This means, a list of tracks at current time t from each sensor processing. Then, the fusion process must fuse these lists of tracks to get a combined list of tracks. As in fusion at detection level, tracking level fusion has to solve the association problem between lists of tracks and implement a mechanism to combine the objects properties from the different lists, e.g., kinetic and class properties. By using an effective fusion strategy at this level, false tracks can be reduced.

The works presented in (Blanc *et al.*, 2004) and (Floudas *et al.*, 2007) described a track level fusion architecture for radar, infrared and lidar sensors. The core idea of these two works consists of defining an association technique to perform track-to-track associations. Blanc *et al.* (2004) proposed a track dissimilarity computation based on gating areas. Floudas *et al.* (2007) formulated the data association problem in presence of multi-point objects and then proposed a solution based on multidimensional assignment. In both cases no information from the class of tracks is included. Moreover, the application is focused on the detection of vehicles on highway scenarios.

Tango (2008) has presented a multiple object tracking data fusion technique between radar data, camera images and ego vehicle odometry. The fusion approach goal

is to obtain more reliable and stable tracks. Besides, the aforementioned method is used to detect stationary objects. This method includes an image processing module to detect vehicles based on line features. However, in low-contrast scenarios, the performance of the vehicle classification based only on line detection decreases.

Fayad and Cherfaoui (2008) proposed an evidential fusion approach to combine and update detection and recognition confidences in a multi-sensor pedestrian tracking systems. Their proposed method is tested using synthetic data. Their results showed that the consideration of reliability in the sources and confidence factors improve the object detection rate.

Gidel *et al.* (2009) presented a multi-sensor pedestrian detection system. The centralized fusion algorithm is applied in a Bayesian framework. The main contributions consist of the development of a non parametric data association technique based on machine learning kernel methods. The performance of the methods depends on the number of particles. Besides, an extension to include other moving objects (e.g., car and truck) makes this method infeasible for real-time applications.

Recently, Baig (2012) has proposed a fusion approach which takes as inputs list of tracks from lidar and stereo-vision sensors. The fusion method uses a gating approach based on the covariance matrices of the tracks for both sensors by calculating statistical distances. However, the aforementioned track association scheme excludes the appearance information from camera images, which could lead to a better object detection and a more accurate object classification.

2.6 Summary

In this chapter we have described the problems of SLAM and DATMO that were briefly introduced in Chapter 1. We have formalized both problems because of their strong correlation to solve the problem of vehicle perception (Wang *et al.*, 2007). Nevertheless, we focused on the DATMO problem to review the state-of-the-art works because our contributions are placed in this component.

We defined each module within the DATMO element to describe the single-sensor architecture of a DATMO solution: moving object detection and moving object tracking. Moreover, we also analysed several solutions for inner problems: data association and filtering.

In an attempt to highlight the importance of object classification we reviewed several state-of-the-art approaches to classify sensor observations. Since we are proposing

a solution for intelligent driving systems; we focus on the classification of vehicles (car and truck), pedestrians and bikes. Following the results obtained by the reviewed works, we decided not only to use camera-based classification but to extract class information from lidar and radar measurements. Regarding camera-based classification, we focus on an appearance based approach due in part to its ability to represent intra-class variety and to the extensive work on feature descriptors and object classifiers.

We presented the related research works based on the fusion techniques that are usually employed to combine information from different sources. Advantages and disadvantages of the fusion techniques were analysed to support our decision to choose Evidential theory as a promising alternative for our application. The advantages are: its ability to represent incomplete evidence; manages conflict; includes reliability; and takes into account uncertainty from the evidence sources; also, its built-in mechanisms to combine evidence. The analysis of the state-of-the-art works was done not only based on the fusion techniques but it included related works based on the level where the fusion is performed. This alternative revision of the state-of-the-art allowed us to focus on final implementations of the fusion methods and mentioned their drawbacks while highlighting the perspectives that motivated our two proposed fusion approaches. Information fusion approaches inside the DATMO component pursue the improvement of the final result of perception task. One focuses on the reduction of mis-detections, the other focuses on the reduction of false tracks. We believe that an appropriate fusion at detection level can achieve both goals while keeping the updating mechanisms of the tracking level fusion.

Also, we have presented the methods we used to process the measurements delivered by three sensors: lidar, radar and camera. These sensor processing approaches represent the first stage of our whole perception solution that will be described in detail in Chapter 3. The results of the sensor processing modules are used by each of our two proposed fusion approaches to extract information from the object detections and build the composite object representation at tracking and detection level. This information extraction process is detailed in Chapters 4 and 5, respectively.

In the following chapters we define our multi-sensor fusion contributions at tracking and detection level. We will also detail the implementations of the different modules inside each proposed fusion approach. Experimental results will follow each fusion approach to show their performance.

CHAPTER 3

Methodology overview

SENSOR configuration, an overview of the proposed fusion architectures, and the sensor data processing are the main topics of this chapter. First, we will introduce the sensor configuration we use to implement and test our different fusion contributions. The sensors presented in this chapter represent the main measurements inputs of the proposed fusion approaches at tracking and detection level detailed in Chapters 4 and 5. The sensor configuration is deployed in a real vehicle demonstrator which forms part of the *interactIVe* project. Hence, we present in this chapter an overview of the interactIVe project which aims at building a perception system as the keystone of a real perception application. This overview gives us the general picture of the goals and requirements of a real perception system and helps us to visualise the inputs and outputs of our proposed fusion approaches. Our implementation of the final real perception system inside the interactIVe project is presented in Chapter 6 following the requirements of the *interactIVe* project presented in this chapter and using the same sensor configuration. Although we are covering in detail our two proposed fusion architectures in the next chapters using a specific set of sensors, in this chapter we introduce the two generic fusion approaches and give and overview of what is expected to be covered afterwards.

Sensor data processing is the first step in every perception system. It involves the sensor configuration, data gathering and data processing. In this chapter, we also present the sensor processing techniques we use to extract useful information from the environment. These techniques focus on the early stages of the perception problem that are usually part of the SLAM component. In Chapters 4 and 5, we will use the representations and data processing methods presented in this chapter to detect, track, and classify moving objects at the respective tracking and detection levels.

Although our main contributions are focused on the DATMO component and pre-

sented in Chapters 4 and 5, in this chapter we show how we apply state-of-the-art approaches to build a SLAM solution extracting information from three different sensors: lidar, radar, and camera. The data processing methods presented in this chapter are common to our two multi-sensor fusion approaches and are also used in the final perception system implementation detailed in Chapter 6.

3.1 interactIVe project

Numerous accident statistics and in-depth studies carried out over the years yield a very uniform picture of road traffic accident causation. Human error as almost as a sole principal causative factor in traffic accidents, has been quoted repeatedly for decades. The limitations of road users are well known and recognized. The *interactIVe* project is addressing this problem by developing next-generation safety systems. These systems are able to compensate driver errors and avoid accidents or mitigate the consequences of a collision, with a focus on active interventions. Therefore, the project belongs to the family of Intelligent Vehicle projects, aiming to deploy advanced technologies for safer and cleaner traffic. These goals have been set by the European Commission, numerous member states, and different stakeholders separately.

Currently available systems are typically implemented as independent functions. This results in multiple expensive sensors and unnecessary redundancy, limiting their scope to premium-class vehicles. The project is based on the idea that by integrating applications together, drivers can be supported more effectively in a larger range of scenarios; moreover, vehicle components may be shared among the various safety systems. This approach, allowing an affordable and robust perception complemented by intelligent intervention strategies, is a key enabler for multiple applications and ultimately can lead to a single safety system well adapted to market introduction at acceptable costs. The vision of *interactIVe* project is accident-free traffic realised by means of affordable integrated safety systems penetrating all vehicle classes, and thus accelerating the safety of road transport.

The general objective of the *interactIVe* project is to develop new high performance and integrated ADAS applications, enhancing the intelligence of vehicles and promoting safer and more efficient driving. Specifically, the project aims to design, develop, and evaluate three groups of functions: Continuous Driver Support; Collision Avoidance; and Collision Mitigation to be introduced in dedicated demonstrator vehicles as shown in Figure 3.1. These vehicles are six passenger cars of different classes and one truck for long-distance delivery.

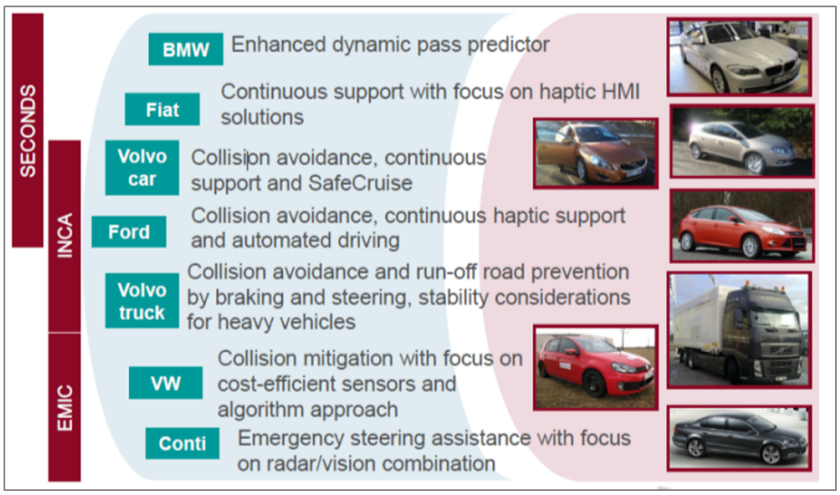


Figure 3.1: Objectives per vehicle demonstrators involved in the *interactIVe* project.

The overall concept of *interactIVe* includes safety functions addressing all the different degrees of hazard, from normal driving to crash scenarios. The functions rely on the data elaborated by the perception layer, and embedded IWI (Information, Warning, and Intervention) strategies, in order to support drivers by warning, active braking, and steering whenever necessary, providing responses always aligned with their expectations.

The number of ADAS applications is growing rapidly on the market today. It is obvious that the number of sensors cannot be increased in the same way as the number of applications is increasing. Instead, sensors of different technologies have to be combined by using sensor data fusion algorithms. The ideal situation would be to use only two to three different types of sensors in vehicles while using robust sensor fusion algorithms for safety applications.

A strong paradigm in the development of Intelligent Vehicle Systems (IVS) is sensor data fusion. Another trend is to use information given by maps and communication systems, aimed at providing drivers with more time to respond to sudden changes in the travel environment, so called foresighted driving. Both approaches assume extensive environment monitoring, data collection, and a perceptual model of the environment to be further used for various safety functions. The project is also based on the concept that by integrating applications together, vehicle components may be shared among the various safety systems. The objective is also to use existing and up-coming sensors, not to develop new ones.

Since *interactIVe* focuses on safety functions, the great majority of the target scenarios have been derived from road accident data. This involves both high-level statistics

(frequency and injury distributions) on the general targeted accident types as well as more detailed descriptions, based on in-depth accident analysis, and on the flow of events (including driver- and vehicle kinematic states) leading to the accident.

The system architecture developed by *interactIVe* comprises discrete architectural layers that are common to all applications. In particular, a modular framework has been defined, based on the following four layers: the sensor layer, the perception layer, the application layer, and what the driver perceives as the system, the IWI layer. Figure 3.2 illustrates the four layers inside the *interactIVe* architecture. All demonstrator vehicles use the same basic architecture but the actual implementation differs. For instance not all demonstrators have the same sensors or the same actuators. The sensor layer (to the left in the figure) transmits all available input data to the perception layer which performs the low level and high level data fusion. Then the processed information derived by the perception modules is transferred to the application layer through the Perception Platform (PP) output interface, namely the Perception Horizon. The application layer performs situation analysis and action planning resulting in a system with different assistance levels, ranging from pure information and warning via gentle activation of different actuators, to full takeover of control, depending on the real-time driving environment.

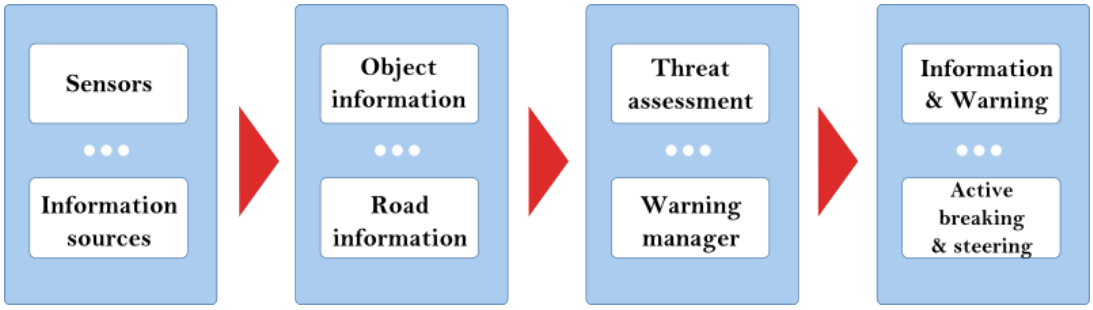


Figure 3.2: Layers inside the system architecture.

Figure 3.3 below illustrates the generic *interactIVe* architecture. The sensor layer transmits all available input data to the perception layer. During input data collection, the data are stored and acquire a common time stamp by the platform Input Manager subsystem. Later on, the various software modules of the Perception Layer process and fuse the sensor input and deliver their unified environment perception output to the applications via a unified interface, the Perception Horizon (PH). The application layer performs a situation analysis and planning resulting in a system with a more energetic (different actuators are activated) or more passive (the system informs and warns the driver) driver assistant role depending on the real-time driving environment.

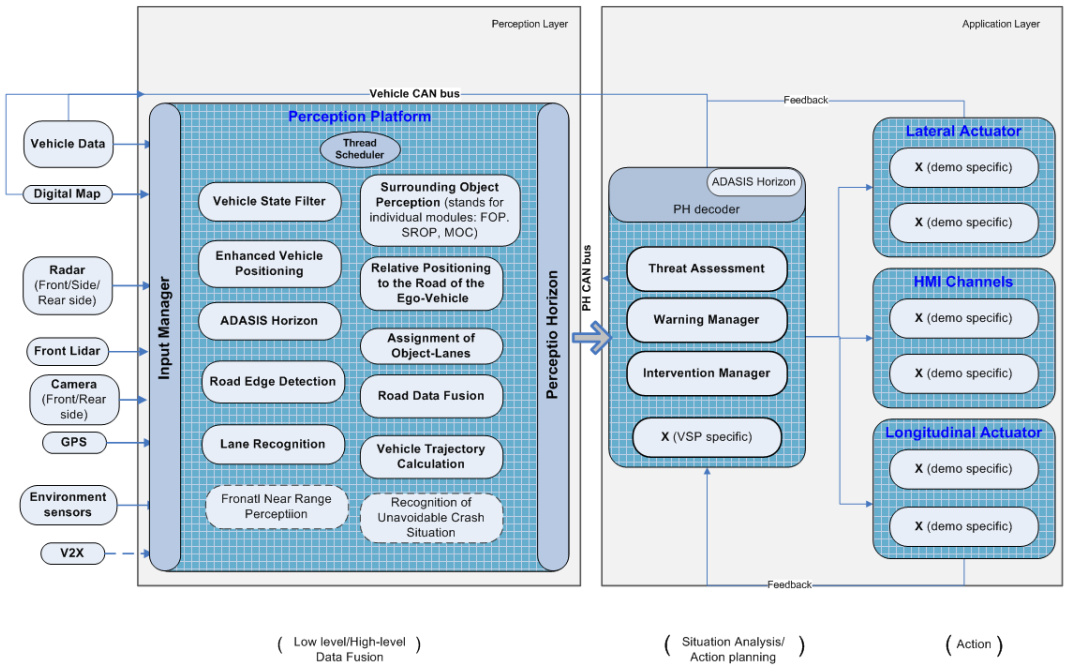


Figure 3.3: *interactIVe* generic architecture.

3.2 Perception subsystem

The perception research area within *interactIVe* heads to construct a unified access point for multiple ADAS applications, which is known as the perception layer. Therefore, not only will different fusion approaches fit into the same concept, but also all applications will access sensor, digital map, and communication data through a common interface: the Perception Horizon.

In the *interactIVe* project, the role of sensing and interpreting the environment is performed by the Perception subsystem. As input to the perception subsystem, different types of sensors are used ranging from radars, cameras, and lidars, to GPS receivers for the extraction of the electronic horizon. The perception subsystem feeds the application layer with a real-time representation of the driving environment, called the Perception Horizon, and thus enables decision-making capabilities on the applications' side. Our research work in the *interactIVe* project takes place inside the perception subsystem and focuses on three main goals:

- To improve the efficiency and quality of sensor data fusion and processing, by investigating and applying our proposed fusion approaches, especially on object detection and classification refinement.
- To support the situation assessment, application modules are built and linked

upon the unified perception platform composed of advanced environment perception modules that should cover a wide range of scenarios.

- To define and develop methodology and evaluation tools for conducting performance assessment of perception research modules in cooperation with the vehicle demonstrator test-case scenarios.

The first goal is covered in Chapters 4 and 5 where we propose two fusion architectures that improve the performance of a multi-sensor fusion set-up. The last two goals are more related with the final perception application detailed in Chapter 6. In this application, we implement the modules developed in our fusion architectures at detection and tracking level complying with the industrial constraints from the *interactIVe* project.

3.2.1 Perception architecture inside the *interactIVe* project

In the *interactIVe* project, multiple integrated functions are developed for continuous driver support, but also for executing active interventions for collision avoidance and collision mitigation purposes are served by an unified perception layer. As was mentioned above, the primary objective is to extend the ADAS scenarios range and usability by introducing a unique access point, the so-called perception layer. Figure 3.4 shows the schematic of the perception platform developed inside the Perception subsystem, which plays the role of sensing and interpreting the environment. The perception subsystem consists of: the sensor interfaces; the perception modules; and the output interface (Perception Horizon). Our implementation of the two modules highlighted by red rectangles is presented in Chapter 6.

3.3 Vehicle demonstrator and sensor configuration

The basic concept behind the development of *interactIVe* functions, and particularly for Continuous Support functions is as if there was an assistant sitting next to the driver. The assistant would be usually silent but can give driving advices to the driver or, in the most dangerous cases, even take control of the car. When the danger is passed, the control is handed over back to the driver.

We used the CRF (Fiat Research Center) vehicle demonstrator, which is part of the *interactIVe* European project, to obtain datasets from highways and cluttered urban scenarios. The datasets obtained were used to test our proposed fusion frameworks. In

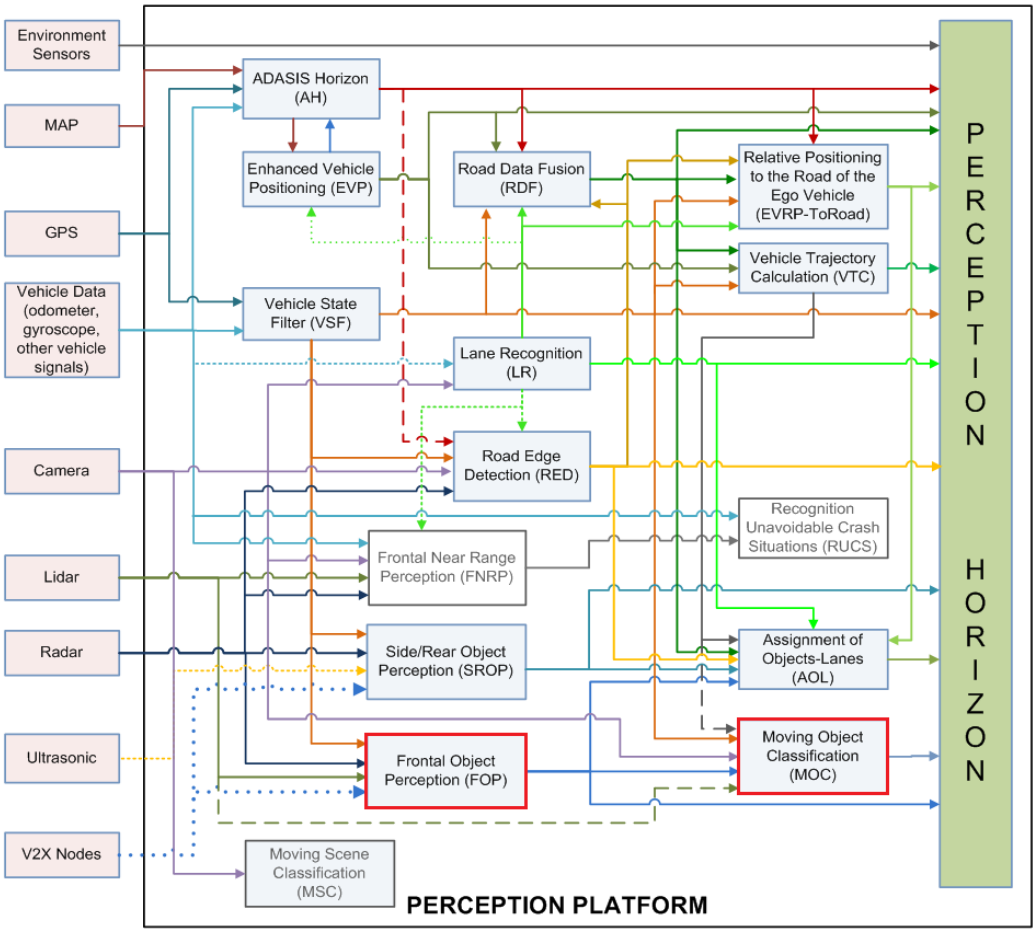


Figure 3.4: Schematic of the perception platform.

order to accomplish the Continuous Support functionalities, the CRF vehicle demonstrator (Lancia Delta car) is equipped with a set of sensors, other information sources, processing units, and driver interaction components as indicated in Figures 3.5 and 3.6. Also, we use gather datasets from real driving scenarios using the sensor configuration inside the CRF vehicle demonstrator to test our two main fusion architectures detailed in Chapters 4 and 5.

The use cases covered by the Continuous Support function implemented on the CRF demonstrator vehicle are summarised in Table 3.1. Also, in dangerous situations, sounds and active feedbacks are activated in the steering wheel and modify the pressure in the safety belts.

The proposed intelligent system application has been tested and evaluated for the need of the project in the CRF demonstrator vehicle and thus it makes use of the CRF front-facing sensor set, which we can see in Figure 3.7. The CRF demonstrator is equipped with the following set of sensors that provide data inputs to the FOP & MOC modules:

Table 3.1: Implemented use cases on the CRF vehicle demonstrator for Continuous Support functions.

Use case description	Visual feedback
Normal situation and unintended lane departure with no side obstacle. In unintended lane departure situation soft feedback on the steering wheel is provided.	
Exceeding speed limits and drift to side barrier. In drift to side barrier situation haptic feedback on the steering wheel and acoustical alarm are generated.	
Vehicle in blind spot (pre-warning and imminent). Imminent warning is generated when lane drift occurs with side vehicle and is associated with haptic feedback on the steering wheel.	
Collision with vulnerable road user and rear end collision (pre-warning and imminent). Imminent warning is associated with acoustic alarm and haptic feedback on safety belt.	
Approaching a curve at high speed (pre-warning and imminent). Imminent warning is associated with acoustic alarm and haptic feedback on safety belt.	

Figure 3.5: CRF demonstrator vehicle on Lancia Delta, with location of sensors and driver-interaction channels. Active safety belt is not shown.

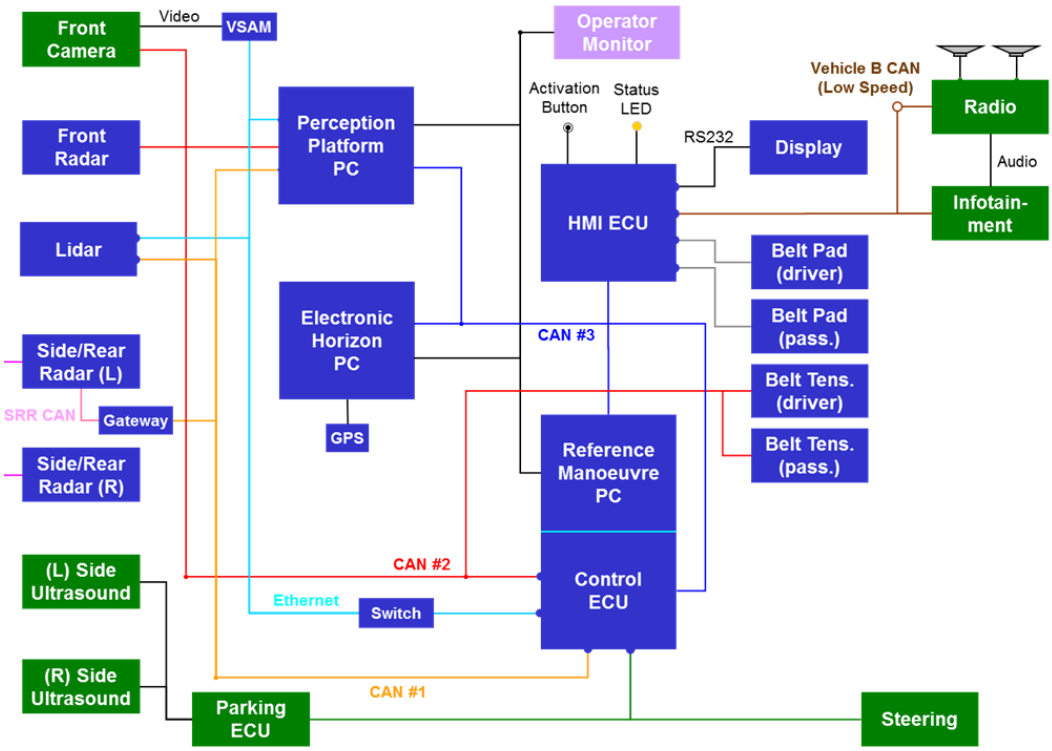


Figure 3.6: Architecture of the system modules for the CRF demonstrator vehicle.

- TRW TCAM+ camera: provides camera images and road lane information. This camera gathers B&W images and has a field of view of $\pm 21°$.

- TRW AC100 medium range radar: provides information about targets detected by the radar sensor. This sensor has a detection range up to 150m, its velocity range is up to 250kph, its field of view is $\pm 12°$ (close range) or $\pm 8°$ (medium range), and its angular accuracy is $0.5°$.

- IBEO Lux laser scanner (lidar): provides a 2D scan of the environment in the form of a list of its impact points to obstacles within its field of view. This lidar has a range up to 200m with an angular and distance resolution of $0.125°$ and 4cm respectively, its field of view is $110°$.

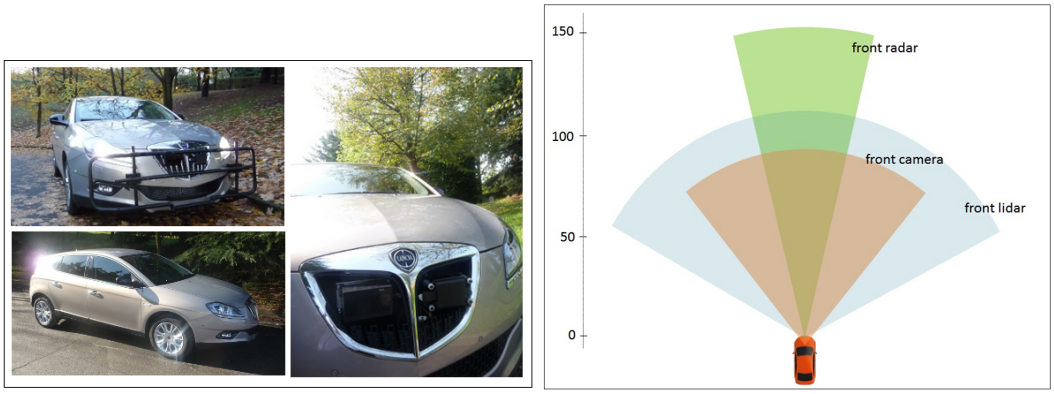


Figure 3.7: Left: images of the CRF vehicle demonstrator. Right: Field of view of the three frontal sensors used as inputs to gather datasets for our proposed fusion approaches detailed in Chapter 4 and Chapter 5, and for the implementation of a real perception application presented in Chapter 6.

3.4 Fusion architectures

The revision of the state-of-the-art in Chapter 2 allowed us to localize the stages where fusion can be performed inside the DATMO component. Also, in Chapter 2, we could analyse the importance of considering class information inside the perception task and not only as an aggregate to the final output. Although class information was not considered a fundamental output for early perception solutions, we consider it a complementary factor that, among kinematic and shape information, can provide a better understanding of the objects of interest present in the scene. Moreover, in the same chapter, we reviewed and highlighted the relevance of the management of incomplete information coming from sensors, data processing and fusion techniques. The reasons stated above, motivated the design of our two main fusion architectures and their modules. We decided to focus on two specific fusion levels: tracking level and detection level.

The fusion at tracking level represents the classical fusion approach followed by most of the fusion solutions inside the intelligent vehicles field (Baig, 2012). Figure 3.8 a) shows the general architecture of our fusion approach at tracking level taking into account a generic number of sensor inputs. In this architecture, object detection and tracking are performed prior to the fusion technique. After tracking, class information is extracted to enrich the object representations in the single-sensor list of tracks. Afterwards, fusion is performed between the single-sensor list of tracks. Although we follow a classical fusion architecture, as an enhancement, we include the class information in this level to enrich the final object representation and to improve the tracking of the moving objects. Moreover, using an evidential framework, we combine the available information of the object tracks and the uncertainty factors from sensor measurements. By doing this, we show how the addition of class information can improve the final results while following a classical architecture.

The fusion at detection level proposes an early inclusion of the available information to improve the moving object detection and tracking while providing and updating a complete object representation. Figure 3.8 b) shows the general architecture of our fusion approach at detection level for a generic number of sensor inputs. For each involved sensor, moving object detection and classification are performed prior to the fusion process. Here, class information is extracted from the object detections provided by each single-sensor object detection module. Tracking is performed using the final list of object representations obtained from the fusion of object detections. The object representation proposed for this fusion architecture includes not only kinematic data but also appearance information about the geometry and class of the objects of interest. Here, the evidential framework powers the fusion process, and also a proposed data association technique to decide which object detections are related.

In Chapters 4 and 5, we show two specific architectures for our fusion approaches at tracking and detection level considering three different sensor inputs: lidar, radar and camera. In these chapters, we also describe specific modules for extracting object information and performing DATMO related tasks. Both fusion approaches at tracking and detection level have to combine the available information from the sensors, and to manage and include the incomplete information into the object representation. Both architectures are powered by an evidential framework which allows to perform the tasks stated above. In the next chapters, this framework is described according to its location inside the respective fusion architectures.

In the next section, we will describe the sensor processing modules that are common to both of our fusion architectures. These modules include the SLAM solution,

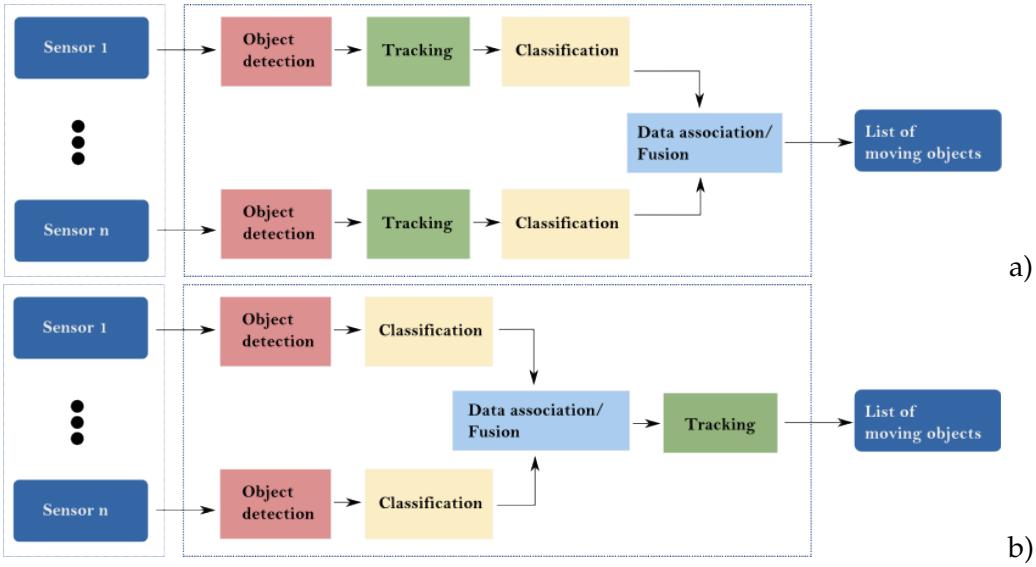


Figure 3.8: General overview of our two proposed fusion architectures at tracking (a) and at detection level (b).

lidar-based moving object detection, radar targets description, image-based representation, and image-based classification. Later on, in Chapters 4 and 5, we will describe some modifications to these modules according to their position inside the fusion architectures.

3.5 Sensor data processing

In this section we describe the sensor processing approaches we used in order to extract useful information from the raw data provided by the three different sensors. These approaches are common to our two different multi-sensor fusion contributions, but for clarity sake we present them at this early stage of the dissertation. We use three different sensors: lidar, radar and camera. Therefore, in the next subsections we will describe how each sensor output is processed to obtain the needed inputs for the next perception components.

3.5.1 Lidar processing

We consider the LIDAR (LIght Detection And Ranging) scanner as the main sensor on our sensor configuration due to the its high resolution and accuracy to detect obstacles. Moreover, it powers the SLAM component of our perception solution. The lidar sensor used in this dissertation is a Lux scanning lidar supplied by IBEO. The lidar scanner is placed on the front bumper of the vehicle demonstrator. The main goal of the lidar is

to have precise measurements of the shape of the obstacles in front of the vehicle. In this dissertation we analyse how much extra information can be derived from such a sensor, e.g. classification information.

As was mentioned in Section 1.2.1, sensor model and inverse sensor model are two important concepts when raw sensor measurements are used. In the case of lidar sensor, the sensor and inverse sensor model are shown in Figure 3.9.

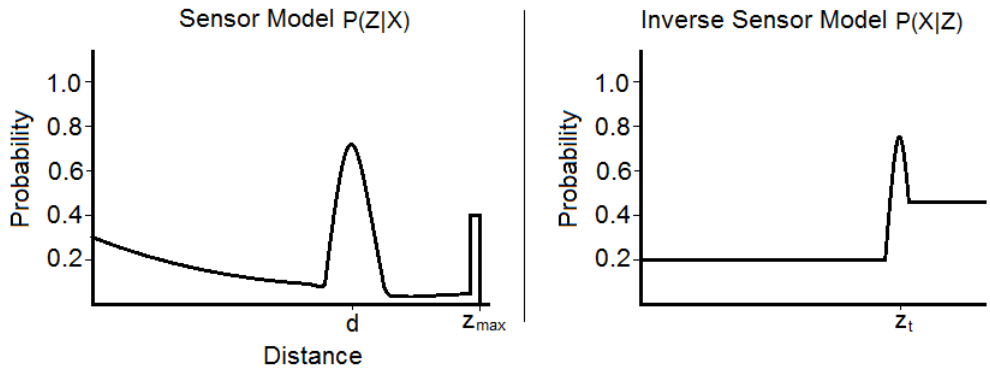


Figure 3.9: *Sensor model* of a laser scanner, the curve models the probability of getting at distance d from the lidar position, z_{max} represents the maximum range of the scanner (left). *Inverse sensor model*, this curve models the probability of the presence of an object at different distances for which laser gives the measure z_t. (right)

Lidar data precision makes it suitable to work during day and night scenarios and is not affected by some weather conditions. These advantages make lidar a preferred choice for modern intelligent vehicle perception. However, classification information provided by lidar is far from being accurate. Moreover, if resolution is not high enough or sensor noise is present, lidar sensor delivers many false positives or mis-detections, making the overall perception results very poor. Fortunately, the deployment of a multi-sensor infrastructure can help overcome these shortcomings.

SLAM component solution

Although our main contributions are focused on the DATMO component, we need to solve the SLAM component in order to obtain the environment map and information about the vehicle localization which ultimately will allow us to extract moving object detections. We use the approach proposed in (Vu, 2009) as a basis for our SLAM solution. Following the same idea, we employ the lidar data as the main source for populating the occupancy grid. Therefore, the output from the lidar data processing described in this section is taken as the input for the SLAM approach proposed below.

As was mentioned in Section 2.2, in an occupancy grid based environment representation, the environment is divided into two dimensional lattice M of fixed sized cells m_i, where each cell has a probabilistic estimate of its occupancy state. For our SLAM solution we used a rectangular shaped cell for our grid based representation. At any point in time the state of the grid represents the map of the environment. In this section, we first present the solution of the mapping part assuming that the position of the vehicle is known (i.e. x_t is known), and then we will discuss how to find the best localization given the updated map.

Formally, the goal of the mapping process is to estimate the posterior probability $P(M|x_{1:t}, z_{1:t})$, where $x_{1:t} = \{x_1, \ldots, x_t\}$ is the vehicle trajectory and $z_{1:t} = \{z_1, \ldots, z_t\}$ are the sensor inputs obtained from start up to time t. If we apply Bayes theorem, this probability is expanded to a form where it is necessary to calculate the marginal probability over the map M for each new sensor input. Periodic calculations make this mapping solution intractable. Works like the ones described by Vu (2009) and Wang *et al.* (2007) have assumed conditional independence of the map cells reducing above probability to $P(m|x_{1:t}, z_{1:t})$, where mapping is equivalent to estimating occupancy state of all individual cells. Therefore, using Bayes theorem we can write our solution as:

$$P(m|x_{1:t}, z_{1:t}) = \frac{P(z_t|x_{1:t}, z_{1:t-1}, m)P(m|x_{1:t}, z_{1:t-1})}{P(z_t|x_{1:t}, z_{1:t-1})}, \tag{3.5.1}$$

Assuming that current sensor measurement z_t does not depend on previous measurements ($z_{1:t-1}$) and previous vehicle positions ($x_{1:t-1}$). We can rewrite the factor $P(z_t|x_{1:t}, z_{1:t-1}, m)$ to $P(z_t|x_t, m)$ so that Equation 3.5.1 becomes:

$$P(m|x_{1:t}, z_{1:t}) = \frac{P(z_t|x_t, m)P(m|x_{1:t}, z_{1:t-1})}{P(z_t|x_{1:t}, z_{1:t-1})}. \tag{3.5.2}$$

If we apply Bayes theorem on $P(z_t|x_t, m)$ Equation 3.5.2 is rewritten as follows:

$$P(m|x_{1:t}, z_{1:t}) = \frac{P(m|z_t, x_t)P(z_t|x_t)P(m|x_{1:t}, z_{1:t-1})}{P(m|x_t)P(z_t|x_{1:t}, z_{1:t-1})}, \tag{3.5.3}$$

where $P(m|x_t)$ represents the prior probability of occupancy of the cell m. Given that $P(m|x_t)$ does not depend on the current vehicle position, Equation 3.5.3 becomes:

$$P(m|x_{1:t}, z_{1:t}) = \frac{P(m|z_t, x_t)P(z_t|x_t)P(m|x_{1:t}, z_{1:t-1})}{P(m)P(z_t|x_{1:t}, z_{1:t-1})}. \tag{3.5.4}$$

After expanding the term $P(m|x_{1:t}, z_{1:t-1})$ using Bayes theorem, Equation 3.5.4 becomes:

$$P(m|x_{1:t}, z_{1:t}) = \frac{P(m|z_t, x_t)P(z_t|x_t)P(x_t|x_{1:t-1}, m, z_{1:t-1})P(m|x_{1:t-1}, z_{1:t-1})}{P(m)P(z_t|x_{1:t}, z_{1:t-1})P(x_t|x_{1:t-1}, z_{1:t-1})}. \tag{3.5.5}$$

Whilst Equation 3.5.5 calculates the probability of cell m being occupied; the probability that cell m is free, $P(\overline{m}) = 1 - P(m)$, is obtained by:

$$P(\overline{m}|x_{1:t}, z_{1:t}) = \frac{P(\overline{m}|z_t, x_t)P(z_t|x_t)P(x_t|x_{1:t-1}, \overline{m}, z_{1:t-1})P(\overline{m}|x_{1:t-1}, z_{1:t-1})}{P(\overline{m})P(z_t|x_{1:t}, z_{1:t-1})P(x_t|x_{1:t-1}, z_{1:t-1})}, \tag{3.5.6}$$

after dividing the probability of cell m being occupied by the probability of cell m being free we get:

$$\frac{P(m|x_{1:t}, z_{1:t})}{P(\overline{m}|x_{1:t}, z_{1:t})} = \frac{P(m|z_t, x_t)}{P(\overline{m}|z_t, x_t)} \frac{P(\overline{m})}{P(m)} \frac{P(x_t|x_{1:t-1}, m, z_{1:t-1})}{P(x_t|x_{1:t-1}, \overline{m}, z_{1:t-1})} \frac{P(m|x_{1:t-1}, z_{1:t-1})}{P(\overline{m}|x_{1:t-1}, z_{1:t-1})}. \tag{3.5.7}$$

Regarding the position of the vehicle, we can state the SLAM problem in two ways. First, if we suppose that current position x_t is known, then Equation 3.5.7 is rewritten as:

$$\frac{P(m|x_{1:t}, z_{1:t})}{P(\overline{m}|x_{1:t}, z_{1:t})} = \frac{P(m|z_t, x_t)}{P(\overline{m}|z_t, x_t)} \frac{P(\overline{m})}{P(m)} \frac{P(m|x_{1:t-1}, z_{1:t-1})}{P(\overline{m}|x_{1:t-1}, z_{1:t-1})}. \tag{3.5.8}$$

For the sake of simplicity, let us define:

$$Odds(x) = \frac{P(x)}{P(\overline{x})} = \frac{P(x)}{1 - P(x)}. \tag{3.5.9}$$

By including Equation 3.5.9 in Equation 3.5.8, we get:

$$Odds(m|x_{1:t}, z_{1:t}) = Odds(m|z_t, x_t) Odds(m)^{-1} Odds(m|x_{1:t}, z_{1:t-1}). \tag{3.5.10}$$

In order to avoid under-flowed results by dividing small probability quantities, Equation 3.5.10 is rewritten using the *log* form as follows:

$$\log Odds(m|x_{1:t}, z_{1:t}) = \log Odds(m|z_t, x_t) - \log Odds(m) + \log Odds(m|x_{1:t-1}, z_{1:t-1}), \tag{3.5.11}$$

where $P(m)$ and $P(\overline{m})$ represent the prior probabilities of a cell being occupied and free, respectively. Commonly, their initial values are set to 0.5 since no information is available. Therefore, as these values are equal, Equation 3.5.11 becomes:

$$\log Odds(m|x_{1:t}, z_{1:t}) = \log Odds(m|z_t, x_t) + \log Odds(m|x_{1:t-1}, z_{1:t-1}), \tag{3.5.12}$$

which finally gives us the calculation of $P(m|x_{1:t}, z_{1:t})$:

$$P(m|x_{1:t}, z_{1:t}) = \left[1 + \frac{1 - P(m|x_t, z_t)}{P(m|x_t, z_t)} \frac{P(m)}{1 - P(m)} \frac{1 - P(m|x_{1:t-1, z_{1:t-1}})}{P(m|x_{1:t-1}, z_{1:t-1})}\right]^{-1}, \tag{3.5.13}$$

where term $P(m|x_t, z_t)$ is known as the inverse sensor model and was defined in Section 3.5.1. One can notice that Equation 3.5.13 can be used to update the occupancy of the cells using different sensor models.

Let us explore the scenario where we no longer know the current position (x_t) of the vehicle. This means, simultaneously estimating the vehicle position in the discovered map. In order to do so, we use an incremental maximum likelihood SLAM

approach (Vu, 2009, Wang *et al.*, 2007). According to this approach the current vehicle pose estimate $\hat{x}_t$ can be calculated by maximizing the marginal likelihood given as:

$$\hat{x}_t = \arg\max_{x_t} \{P(z_t|x_t, \hat{M}_{t-1})P(x_t|u_t, \hat{x}_{t-1})\}, \tag{3.5.14}$$

where $\hat{M}_{t-1}$ is the map at time $t-1$ and $\hat{x}_{t-1}$ is the estimated previous pose. Once we have determined the pose of the vehicle, we update the occupancy grid map using the previous map at time $t-1$ and new incoming sensor measurement using Equation 3.5.13. The process of updating the map is formally represented as:

$$\hat{M}_t = \hat{M}_{t-1} \bigcup \overline{M}, \tag{3.5.15}$$

where $\overline{M}$ is the new map obtained from the estimated pose $\hat{x}_t$ and current measurement z_t, which is also known as *instantaneous* map.

Let us analyse in detail the parts involved in Equation 3.5.14. The term $P(x_t|u_t, \hat{x}_{t-1})$, called motion model, gives the probability of being at position x_t when control u_t is applied from the previous estimated position $\hat{x}_{t-1}$. Term $P(z_t|x_t, \hat{M}_{t-1})$ is the measurement model and gives the probability of getting the measurement z_t from pose x_t and previous estimated map $\hat{M}_{t-1}$. Maximizing Equation 3.5.14 is equivalent to finding the vehicle pose x_t that satisfies the motion model $P(x_t|u_t, \hat{x}_{t-1})$ which best fits measurement z_t.The maximization problem can be reduced to: defining a probabilistic motion model; sampling sufficient values of new pose x_t from this model; and for each value fitting the new measurement z_t to the map $\hat{M}_{t-1}$. The sample of x_t that best fits z_t is taken as the new pose estimation $\hat{x}_t$. For a control input consisting of translational and rotational velocities $u_t = (\nu_t, \omega_t)$ a probabilistic motion model is shown in Figure 3.10.

We can see the term $P(z_t|x_t, \hat{M}_{t-1})$ as the probability that the current measurement z_t fits into the map $\hat{M}_{t-1}$ with respect to the current sampled value of x_t. As we used raw lidar data to perform SLAM, we project the current measurement z_t on the map constructed up to time $t-1$ ($\hat{M}_{t-1}$) taking the current sampled value of x_t as the position of the vehicle. If we define c_i as the cell corresponding to the hit point of *ith* laser beam in z_t and $\hat{M}_{t-1}[c_i]$ as the occupancy probability of cell c_i in the map $\hat{M}_{t-1}$, then the probability $P(z_t|x_t, \hat{M}_{t-1})$ can be computed as:

$$P(z_t|x_t, \hat{M}_{t-1}) \propto \sum_{i}^{N} \hat{M}_{t-1}[c_i], \tag{3.5.16}$$

where N represents the total number of beams in z_t. This approximation is evaluated for different samples of x_t, and the one giving the maximum value is considered the updated estimated pose $\hat{x}_t$.

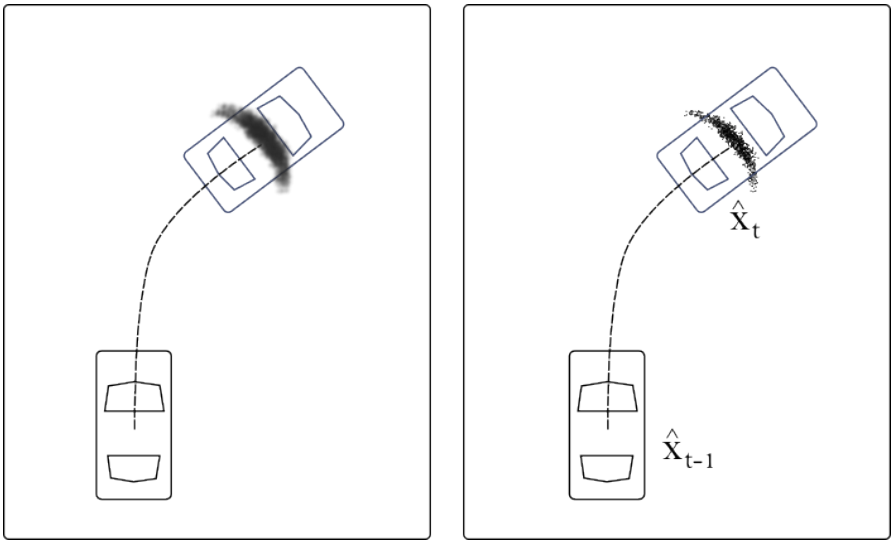


Figure 3.10: Vehicle motion model $P(x_t|u_t, \hat{x}_{t-1})$ (left). Sampling of vehicle motion model (right). Vehicle position is considered to be at the center of the vehicle. Blur gray extremes represent the uncertainty in the two components of control input $u_t = (v_t, \omega_t)$.

The process described above is known as maximum likelihood based SLAM. In brief, it consists of finding the best vehicle pose estimate $\hat{x}_t$ according to the Equation 3.5.14 which needs the definition of a motion model $P(x_t|u_t, \hat{x}_{t-1})$ and a measurement model $P(z_t|x_t, \hat{M}_{t-1})$. Afterwards, this method uses the pose estimate ($\hat{x}_t$) and the latest sensor measurement z_t to update the map following Equation 3.5.15 by applying the cell update process defined in Equation 3.5.13.

Moving Object Detection

When performing SLAM in dynamic environments, such as daily driving scenarios, measurements from sensors, in this case lidar, can belong to static or moving obstacles. In scenarios with several moving obstacles, the localization technique mentioned in the previous section might be affected. Moreover, including measurements from moving objects into the mapping process decreases the quality of the resulting map. These issues motivate the differentiation between measurements from static and moving objects.

Therefore, we will describe our solution for moving object detection based on the SLAM solution proposed in previous section which uses lidar data to populate the occupancy grid. The SLAM algorithm described in Section 3.5.1 incrementally builds a consistent local map of the environment. We follow the assumption that moving ob-

jects can be detected whenever new measurements arrive. The main idea is to identify the inconsistencies between free and occupied cells within the local grid map. The hypothesis is that if an occupied measurement is detected on a location previously set as free, then it belongs to a moving object. Oppositely, if a free measurement is observed on a location previously occupied then it probably belongs to a static object. However, if a measurement is observed in a previously not observed location, then nothing can be said about its moving or static nature.

Additionally, we can use information from previously detected moving objects to improve the detection of moving entities at current time. For example, if there are many moving objects passing through an area, then any object that appears in that area should be considered as potential moving object. This means that we need to store an occupancy grid with the previous detected moving objects. Hence, at all times we have two occupancy grid structures: the local static map M, constructed by the SLAM method described in Section 3.5.1; and a local dynamic grid map D, which stores information of previously detected moving objects. The pose, size, and resolution of the dynamic map is the same as those of the static map. Each dynamic grid cell stores a value indicating the number of observations that a moving object has been observed at that cell location.

Following the previous assumptions, we can divide our moving object detection solution into two steps. First, it detects the measurements that might belong to moving objects. Given a new lidar scan $Z = \{z_1, ..., z_n\}$ (set of lidar beams), the corrected vehicle location x_t, the local static map M built by the SLAM method, and the dynamic map D containing previously detected moving objects, our method classifies a new single measurements z_k as follows:

$$state(z_k) = \begin{cases} static & \text{if } M(k) = \text{ occupied} \\ moving & \text{if } M(k) = \text{ free } or\ D(k) > \alpha \\ uncertain & \text{if } M(k) = \text{ unobserved} \end{cases} \tag{3.5.17}$$

where $M(k)$ and $D(k)$ represent the cell in the static and dynamic grid corresponding to the z_k beam hit. The term α is an uncertainty threshold which helps filter possible false positives.

Second, after the moving measurements are detected, we need to group them into moving objects. In order to do so we use a distance based clustering method. These methods create clusters which represent moving objects. Two moving measurements are considered part of the same moving object if the distance between them is less than β distance threshold. However, the calculation of the distance in the grid is performed

using the lidar resolution. Once the grid D is populated, the clustering process takes the first occupied cell and forms an initial cluster by labelling it. Then, it starts to expand the labelling process to occupied neighbour cells, considering the threshold β. A cell can only be part of one cluster (moving object), therefore it is labelled only once. This process is repeated until no more unlabelled cells remain. Figure 3.11 shows an example of the evolution of the moving object detection process using our two-step occupancy grid method.

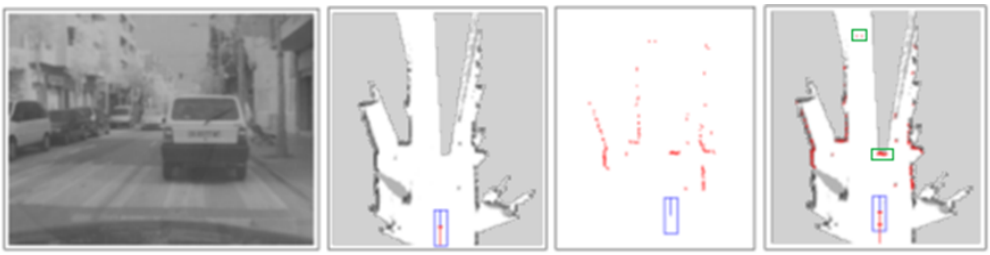


Figure 3.11: Occupancy grid representation obtained by processing raw lidar data. From left to right: Reference image from camera; static occupancy grid M_{t-1} obtained by applying the SLAM solution; current lidar scan represented in an occupancy grid; detection of the moving objects (green bounding boxes).

It is important to notice that once moving objects are detected, the list of objects is taken into account to update the map M. Specifically, measurements detected as parts of a moving object are not used to update the map in SLAM. Measurements marked as uncertain are assumed to be static until latter measurements can support the opposite. This filtering process helps to reduce the number of spurious static obstacles and produces a better quality map M.

3.5.2 Radar targets

The radar sensor provides a list of possible moving objects (also known as targets) inside its field of view. The radar sensor we use is a long range radar sensor model named AC100, supplied by TRW. The radar sensor operates at 24 GHz and is placed in the front bumper of the vehicle demonstrator. Figure 3.7 shows the placement of the radar sensor. Among with the positioning, radar sensor also provides a relative speed estimation of the targets.

The mid-range TRW radar uses internal processing able to detect static and moving obstacles having a radar cross-section similar to a car. The list of these objects, called targets, is delivered as an output to the perception approach. Radar data is given as a list of n targets detected in the radar field of view. Each element of the list includes the

range, azimuth and relative speed of the detected target. As the sensor will produce a return for each object with a significant radar cross section, targets may correspond to static objects or objects other than vehicles, producing false positives. In a similar way, *weak objects* like pedestrians can not always be detected, consequently producing mis-detections. Therefore, it is necessary to address these issues in the next stages of the processing.

3.5.3 Camera images

Camera images are commonly used as reference information for the perception results. In our approaches we use mono-camera sensor to include appearance information into the perception components. We are interested in how early appearance information, such as object class, can improve the object detection and object tracking. We use the high performing sensor camera called TCAM+, provided by TRW. The camera sensor is placed on the front wind-shield of the vehicle demonstrator. Figure 3.7 shows the field of view of the camera sensor along with the other two sensors on the vehicle demonstrator.

Visual representation

As reviewed in Section 2.4, there are several image-based object representations. The selection of the visual representation is related to the type of objects one wants to detect, and to final the application. The HOG descriptor has shown promising results in vehicle and pedestrian detection (Dollár *et al.*, 2012). Therefore, we decided to focus on this descriptor as the core of our vehicle and pedestrian representation.

The camera sensor provides a gray-scale image of the current scenario in front of the vehicle. The goal of object representation is to focus on the possible object of interest and generate a visual descriptor which is used in future stages to determine whether the image contains an object of interest or not. Following the object classification scheme described in Section 2.4, the current section focuses on the hypothesis generation; and in the next section, we will describe the hypothesis verification step.

Histograms of oriented gradients (HOG) are feature descriptors mainly used for object detection purposes. In this work we use them to represent and detect vehicles and pedestrians. A HOG descriptor is a set of feature vectors. Each feature vector is computed from a block placed across a source image. Each element of a vector is a histogram of gradient orientations (Dalal and Triggs, 2005).

The general algorithm of finding the HOG descriptor is described in Algorithm 1.

It takes as input an image and delivers its visual descriptor.

Algorithm 1: General algorithm for HOG descriptor computation

input : Image I of the object of interest
output: HOG descriptor of the given image

```
1  foreach pixel p in the image I do
2      Compute gradients;
3      Perform binning of gradients orientation;
4  end
5  Define a cell size Cs;
6  foreach cell c in the set of all cells C do
7      Collect the histogram within;
8  end
9  Weight the histogram cells for local normalization of the contrasts;
10 Normalize the histogram across the blocks;
```

One can notice that the size of the HOG descriptor strongly depends on the size of the block, cell, and the number of bins. The common HOG computation configuration delivers a descriptor size of over 3 thousand values. Computing HOG descriptors off-line do not represent an issue, but when it comes to on-line processing, this computation time directly affects the final perception application.

We have proposed a possible solution to the high dimension descriptor. We have called Sparse HOG (S-HOG). The main idea is to focus on specific areas of the image we want to describe. For example, when it comes to describe pedestrians we can focus on the upper, lower, and sides of the pedestrian.

We based our visual representation approach on the work of Dalal and Triggs (2005) on histograms of oriented gradients (HOG) which has recently become a state-of-the-art feature in the computer vision domain for object detection tasks. In their original idea, a detection window is divided into a dense grid of cells and histograms of gradients are computed over all overlapping square blocks of the four adjacent cells. From experiments, we discovered that for a given type of object (e.g., pedestrian and vehicle), a block is not necessarily square and by only using a few of the most informative blocks we could represent the object image to obtain similar performance with the benefit of much less computing effort. The resulting feature vector is quite compact when the dimension is about a few hundred compared with about several thousand in the original method.

Figure 3.12 illustrates some of the blocks we selected to extract features for different

object classes: pedestrian, car, truck. It turns out that these selected blocks correspond to meaningful regions of the object image (for example: head, shoulder, legs for pedestrian class). This is the reason why we call it Sparse-HOG. In our implementation, we used 6 histogram bins for all object classes, 9 blocks for pedestrian, 7 blocks for car, and 7 blocks for truck. To accelerate S-HOG feature computation, we employed the idea of using an *integral image* introduced by Viola and Jones (2001b). We compute and store an integral image for each bin of the HOG (resulting in 6 images in our case) and use them to efficiently compute the HOG for any rectangular image region which requires only 4 x 6 image access operations.

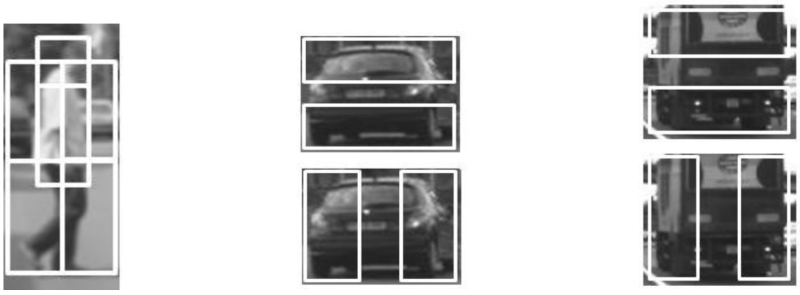


Figure 3.12: Informative blocks for each object class detection window, from left to right: pedestrian, car, and truck (for sake of clarity, only some of them are displayed). Histograms of gradients are computed over these sparse blocks and concatenated to form S-HOG descriptors. Average size of the descriptors for pedestrians, cars and trucks are 216, 288 and 288 respectively.

Object classification

In order to classify objects from camera images, first we follow the most popular approach using a sliding-window paradigm where the detection window is tried at different positions and scales. For each window, visual features are extracted and a classifier (usually pre-trained off-line) is applied to decide if an object of interest is contained inside the window. Once more, the choice of the visual representation and classification method decides the performance of the whole system.

The object representation scheme described in Section 3.5.3 is applied not only for hypothesis generation but as well for hypothesis verification. Moreover, we use a Machine Learning technique to build a classifier using the HOG descriptors computed for each set of objects. This technique needs to have as an input a set of positive and negative examples of the object it intends to classify. In our case, the positive examples are the set of images containing an object of interest (i.e., pedestrian, car, truck);

the negative examples are images without the objects of interest, generally images of clutter.

Once we have computed the descriptors, the choice of classifier has a substantial impact on the resulting speed and quality. To achieve a suitable trade-off, we chose the discrete Adaboost method (Friedman *et al.*, 2000), a boosting-based learning algorithm. The idea of a boosting-based classifier is to combine many weak classifiers to form a powerful one where weak classifiers are only required to perform better than chance hence they can be very simple and fast to compute. Algorithm 2 describes the Adaboost classifier building process we follow to generate our classification models.

Algorithm 2: Adaboost algorithm used to build the classifiers for cars, pedestrians and trucks. Our implementation is based on the work of Freund (1995)

input : $(x_1, y_1)...(x_m, y_m)$ where $x_1 \in X, y_1 \in Y = \{-1, +1\}$
output: A classifier C

```
D_1(i) = 1/m;
for t = 1 to T do
    Train weak classifier with D_t;
    Obtain weak hypothesis h_t : X → {−1, +1} with error ε_t;
    α_t = (1/2) ln((1−ε_t)/ε_t);
    Update ;
    D_{t+1}(i) = (D_{t+1} exp(−α y_i h_t(x_i)) / Z_t) D_t(i);
end
H(x) = sign(Σ_{t=1}^{T} α_t h_t(x));
```

Although there are more options to build a classifier, such as Support Vector machines and Random Forest, we choose a Boosting technique because it combines many weak classifiers to build a powerful one. Moreover, the weak classifiers are only required to perform better than random, and a multi-class classifier can be constructed using binary ones. Regarding Algorithm 2, a weak classifier finds a weak hypothesis $h_t = X \rightarrow \{-1, +1\}$, then the algorithm adapts to the error rates of the individual weak hypothesis. Moreover, this algorithm reduces the training error $\epsilon_t = \sum_i h_t(x_i) \neq y_i$ and maintains a distribution or set of weights over the training set, whilst D_t distribution updating focuses on the classification of the hard samples.

For each object class of interest (e.g., pedestrian, car, truck), a binary classifier was pre-trained to identify object (positive) and non-object (negative) images. For the off-line training stage, positive images were collected from public (such as the Daimler dataset) and manually labelled datasets containing objects of interest from different viewpoints, for example: pedestrian (frontal, profile), car (frontal, rear side), truck

(frontal, rear side). They were all scaled to have sampling images of the same size for each object class: pedestrian: 32x80 pixels, car: 60x48 pixels, truck: 60x60 pixels. Negative samples were generated randomly from images which do not contain an object of interest. S-HOG features were computed for all samples, and then used for the training of classifiers.

The training process starts when each training sample is initially assigned the same weight and iterates for a fixed number of times. In each round, a weak classifier is trained on the weighted training data and its weighted training is recomputed. The weights are increased for training samples being misclassified so that the weak classifier is forced to focus on these hard samples in the next step. The final classifier provides the sign of the weighted sum over individual learned weak classifiers. In our implementation, decision trees are used as weak classifiers powering the boosting scheme.

Final classifiers for each object class (e.g., pedestrian, car, truck) obtained after offline training are used for the on-line object detection stage in a sliding-window scheme. Detection time is affected by both phases of feature extraction and classification. For an input image of 752x250 pixels in size, there are several thousand windows to check and the whole detection time is about 70ms for each object class. Figure 3.13 shows examples of the pedestrian and car detection results (green and red boxes respectively) before merging into the final objects.

In this image-based object classification approach, the confidence of object detection (classification) can be directly estimated based on the number of detection boxes around the object location. Generally, the greater the number of positive windows (containing an object of interest), the greater the confidence that the object belongs to that specific class. Experimentally, we have noticed that false alarms (detections) are often returned with very few positive responses.

3.6 Summary

In this chapter, we presented the sensor configuration used for our two fusion architectures and for the final implementation of the perception system detailed in Chapter 6. Moreover, an introduction of the *interactIVe* project was presented in order to give a general picture of the goals and requirements of a real perception solution. Furthermore, we presented an overview of the proposed fusion architectures at detection and tracking level giving a general description of each one. This description considered a generic number of sensor inputs and illustrated the position of each of the modules

Figure 3.13: Examples of successful detection of pedestrians (top) and cars (bottom) from camera images.

involved in the multi-sensor fusion approaches. Also, we presented the data processing methods for each involved sensor. These methods provide representations of the environment sensed by each sensor. In the case of lidar scanner, we detailed our implementation of a SLAM solution and proposed an approach for moving object detection that is used for the tracking method detailed in the next chapters. Also, this lidar-based representation of the environment is used in Chapters 4 and 5 to extract preliminary shape and class information. We described the data from the targets provided by radar sensor which will be used in the next chapters to represent the position and class of the radar detections; and in the case of Chapter 4 to perform tracking of moving objects. For the camera sensor, we introduced the visual representation we use to generate object hypotheses. The visual descriptor presented in this chapter is the core visual representation for all the modules that process camera images. Our visual hypotheses verification process is performed inside the DATMO component and is explained in the next chapters in Sections 4.1.3 and 5.1.3 where the moving object detection and

classification methods are detailed.

CHAPTER 4

Multi-sensor fusion at tracking level

OUR multi-sensor fusion approach at tracking level is presented in this chapter. We describe the modules inside this fusion approach using information from three main sensors: lidar scanner, mono camera, and radar. Lidar sensor intends to provide precise distance measures between the vehicle demonstrator and the detected objects. Camera sensor gives richer appearance information about the objects, such as classification information. Radar sensor provides a list of targets (unconfirmed detections) and their relative velocity. This sensor configuration aims at building a more elaborate environment model. Performing fusion at tracking level is a common modern approach. Nonetheless, we propose an enhanced fusion approach composed of an updating/combination process and powered by a richer representation of the tracked objects. This representation takes into account preliminary class information from the three different sensors. The goal of our fusion approach at tracking level is to obtain a more accurate list of tracked objects. Where accuracy involves not only less false tracks but also less objects mis-classifications.

Figure 4.1 shows the general architecture of the proposed multi-sensor fusion approach at tracking level for a generic number of sensor inputs. We consider moving objects classification as an important module inside this approach. Therefore, we describe a proposed object state representation which integrates an evidence distribution for the object's class hypotheses. Our fusion approach takes place after the tracking stage of each sensor input. At the end of this chapter, we present experimental results to show the performance of our approach and finalize the chapter with a summary of the reviewed topics.

In order to describe our fusion method, we will follow the architecture in Figure 4.2. This architecture shows in detail the sensor configuration and highlights the early modules involved from the beginning of the DATMO component until the proposed fusion

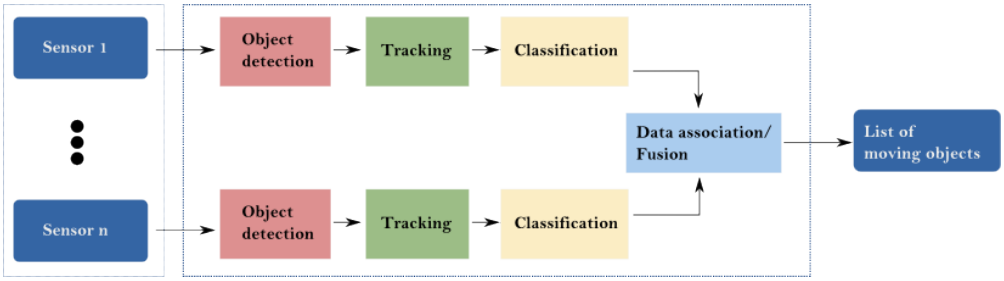


Figure 4.1: General architecture of the multi-sensor fusion approach at tracking level.

module at the end of the component. We can see that a classification step is performed after the tracking process of each sensor. Camera sensor is a special case, due to the lack of accurate data to perform DATMO using only images, we decided to use camera images to extract only appearance information, such as class of objects.

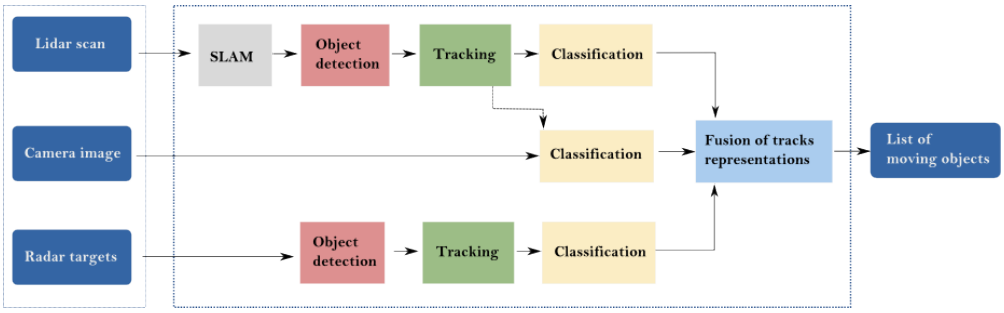


Figure 4.2: Architecture of the multi-sensor fusion approach at tracking level for lidar, camera and radar sensors.

Our hypothesis states that by adding classification information late in the DATMO component we can improve the moving object description and the final output of the perception task. The results obtained by Vu (2009) and Wang *et al.* (2007) motivated the integration of appearance data as complementary information to the tracking process because despite their good results, these works did not consider appearance information in their whole perception solution. By the end of this chapter, we will show how our fusion approach offers a richer output than a single sensor perception approach and improves the single-sensor perception results.

Although we are focusing on the DATMO component, we have already described the SLAM solution in Section 3.5.1. Therefore, from now on we will assume the SLAM component as solved, which means that we already have the SLAM outputs: vehicle localization and local map. Moreover, in Section 3.5.1 we have described the moving object detection process based on lidar sensor data which used the SLAM solution to discriminate moving from static obstacles.

Our multi-sensor fusion approach for the DATMO component proposes an information fusion framework which allows to incorporate in a generic way information

from different sources of evidence. This fusion approach is based on Dempster-Shaper (DS) theory and aims to gather classification information from moving objects identified by several classification modules. These modules can use information from different kinds of sensors. The proposed approach provides a fused list of classified objects as output.

Given a list of detected objects and a preliminary classification from different individual sensor-based classifiers, the proposed approach combines instantaneous information from the current environment state by applying a rule of combination based on the one proposed by Yager (1985). The rule of combination can take into account classification evidence from different sources of evidence (object classifiers), the uncertainty coming from the reliability of the sensors, and the sensor precision to detect certain classes of objects. The proposed approach aims at improving the individual object classification provided by class-specific sensor classifiers and directly improve the final output of the DATMO component. After instantaneous fusion is done, the proposed approach fuses it with the combined results from previous times, which we call *dynamic fusion*. This fusion architecture allows to give more importance to the classification evidence, according to its uncertainty factors.

Before formally describing our fusion approach at tracking level, we first describe our methods to detect and track moving objects using the environment representations described in Chapter 3. We detail the methodology to generate the inputs for our fusion approach from each one of the three sensors. In order to do so, we present two different tracking approaches for lidar and radar. At the same time, we show how we build our composite object representation by extracting classification information from the three sensors. Once we build our object representation for the tracked objects, we perform an evidential fusion based on the class information of the objects and a Bayesian fusion to combine their positions. Afterwards, we present experimental results and an evaluation of the obtained results. The chapter finishes with a summary of our methodology proposal.

4.1 Moving object detection, tracking and classification

In this section, we start to present our proposed solution for the DATMO component. Specifically, we describe the moving object detection process for each sensor and our approaches for extracting classification information. We follow the architecture presented in Figure 4.2 and describe each of the involved modules. The outputs of these modules are taken as inputs for our proposed fusion approach at tracking level which

will be detailed in Section 4.2.

4.1.1 Lidar sensor

Raw lidar scans and vehicle state information are processed to recognize static and moving objects, which will be maintained for tracking purposes. We employ a grid-based fusion approach originally presented in Section 3.5.1, which incrementally integrates discrete lidar scans into a local occupancy grid map representing the environment surrounding the vehicle demonstrator. In this representation, the environment is discretized into a two-dimensional lattice of rectangular cells; each cell is associated with a measure indicating the probability that the cell is occupied by an obstacle or not. A high value of occupancy indicates the cell is occupied and a low value means the cell is free. By using the aforementioned grid-based representation noise and sparseness of raw lidar data can be inherently handled, moreover no data association is required. We analysed each new lidar measurement to determine if static or moving objects are present.

We represent the moving objects of interest by simple geometric models, i.e., a rectangle for vehicles (cars and trucks) and bicycles, and a small circle for pedestrians. Considering simultaneously the detection and tracking of moving objects as a batch optimization problem over a sliding window of a fixed number of data frames, we follow the work proposed by Burlet *et al.* (2007). It interprets the laser measurement sequence by all the possible hypotheses of moving object trajectories over a sliding window of time. Generated object hypotheses are then put into a top-down process (a global view) taking into account all object dynamic models, sensor model, and visibility constraints. A Markov Chain Monte Carlo (MCMC) technique is used to sample the solution space effectively to find the optimal solution and is defined in the following subsection.

Multiple Object Tracking

The background idea behind Markov Chain Monte Carlo (MCMC) method is as follows. A Markov chain can be designed to sample a probability distribution $\pi(\omega) = P(\omega|Z)$. At each iteration, the method samples a new set of possible object tracks ω' from the current state ω_n following a proposal distribution $P(\omega'|\omega_{n-1})$. Where $\omega = \{\tau_1, \tau 2, ..., \tau_K\}$, and τ_k is defined as a sequence of the same object appearing up to current time $\tau_k = \{\tau_k(t_1), ..., \tau_k(t_{|\tau_k|})\}$; here, $|\tau_k|$ is the length of the track τ_k; $\tau_k(t_1)$ represents the moving object detected at time t which is defined by an object state (or

model). In resume, the sampling method estimates the new state generated by the Markov chain from the previous state. The new candidate state ω' is accepted according to the following probability $A(\omega_{n-1}, \omega')$:

$$A(\omega_{n-1}, \omega') = min(1, \frac{\pi(\omega')}{\pi(\omega_{n-1})} \frac{q(\omega_{n-1}|\omega')}{q(\omega'|\omega_{n-1})}), \tag{4.1.1}$$

where if the sample is rejected, the value of ω_{n-1} is kept. $\frac{\pi(\omega')}{\pi(\omega_{n-1})}$ is the relative probability of the states ω_{n-1} and x'. Factor $\frac{q(\omega_{n-1}|\omega')}{q(\omega'|\omega_{n-1})}$ represents the complexity of going from ω' to ω_{n-1} and vice-versa.

If the probability of the proposed state, combined with the probability of going back to ω_{n-1} is greater than the reverse, the sample is accepted. If the candidate state ω' is accepted, ω_n is set to ω'; otherwise, $\omega_n = \omega_{n-1}$. The basis of the algorithm for MCMC we used for the sampling process is described in Algorithm 3. The probability $q(.)$ is known as the dynamics of the Markov chain. There are usually two kinds of dynamics: jump and diffusion. Jump refers to the motion of Markov chain between subspaces of different dimensions and diffusion refers to its motion within a subspace.

Algorithm 3: Sampling algorithm based on MCMC (Roberts, 1996).

```
input  : Z, n_mc, ω* = ω_0
output: ω*

for n = 1 to n_mc do
    Propose ω' according to q(ω'|ω_{n-1});
    Sample U from distribution Uniform[0,1];
    if U < A(ω, ω') then
        ω_n = ω';
        if U < A(ω, ω') then
            ω* = ω_n;
        end
    else
        ω_n = ω_{n-1};
    end
end
```

The choice of the proposal probability $q(.)$ in Algorithm 3 can affect the efficiency of the MCMC significantly. Whilst a random proposal probability will lead to a very slow convergence rate, a proposal probability designed with domain knowledge will make the Markov chain traverse the solution space more efficiently.

The tracking solution we implemented is comprised of two stages. The first one is

composed of the moving object detection process described in Section 4.1.1. Although this process generates many true detections, it can deliver false alarms. Therefore, we will consider the lidar detections as hypotheses of moving objects. These hypotheses are used to generate proposals for a second stage MCMC sampler, with jump/diffusion dynamics, that performs an exhaustive search over the spatio-temporal space of potential moving objects to find the most likely trajectories of moving objects with a maximum posterior probability.

Using the moving clusters from moving object detection the method generates object hypotheses based on predefined object models. The goal is to generate all possible hypotheses corresponding to potential moving objects. For each cluster, a minimum bounding box is computed and the corresponding sides of the cluster are extracted. It is important to notice that at one time instant (sensor scan), a maximum of two sides of a cluster can be seen by the laser sensor. We follow the cluster hypotheses generation described by Vu (2009). Providing that the size of a bounding box of a segment is larger than a threshold, the segment is classified as an L-shape if it has two visible sides, as an I-shape if only one side is visible. Otherwise, it is classified as a mass-point-shape. Depending on the shape and size of clusters, object hypotheses are generated. L-shape clusters generate bus and car hypotheses, I- shape clusters generate bus, car and bike hypotheses, and mass-point clusters generate pedestrian hypotheses. Figure 4.3 shows the fitting model process based on the visibility and size of the clusters.

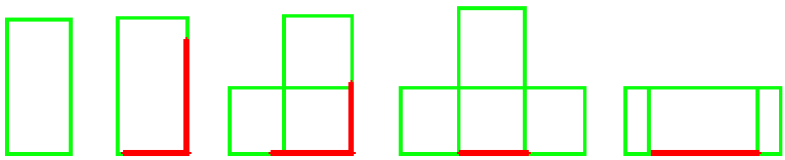


Figure 4.3: Fitting model process: fixed object box model (green); L-shape and I-shape segments (red).

The process of defining the size and shape of the objects is performed using a priori knowledge for each class of object. This knowledge is extracted from real data from road scenarios. Therefore, a fixed rectangular model of 1.7m width by 4.5m length represents the car class. Trucks are represented by a rectangular box of width of 2.5m and length between 9m and 12m. A typical bike length is 2.1m and its width is 0.5m. Pedestrians are represented by a circular shape of 0.5m in diameter.

The object state vector is defined by box model $M = \{x, y, w, l, c, \theta\}$ in case of large object detections. Where x and y are the center of the box, w and l are the width and length according to the class of object c, and θ is the object orientation. For small object

detections a point model $M = \{x, y, c\}$ is used, where (x, y) and c represent the object center and class, respectively.

Once the samples are generated, it is necessary to find the solution of object trajectories from the generated hypotheses space from time window $I = [1, T]$, where T represents the number of frames considered to perform the hypotheses generation process. To do this, we follow the work of Vu (2009), where a neighbourhood graph of hypotheses is generated.

The graph $< V, E >$ represents the relationships of all the moving object hypotheses generated within the time interval I. Let us define h_t^i as the i-th hypothesis generated at time t. Each hypothesis h_t^i is represented by a node in V. The neighbourhood relations between two nodes connected by arcs in E are defined as follows:

- Sibling edges. $E_{sb} = \{(h_{t_1}^i, h_{t_2}^j)\}$, where $t_1 = t_2$ which means that both hypotheses are generated at the same time step. Additionally, h_t^i and h_t^j must share a spatial overlap in the grid representation.
- Parent-child edges. $E_{pc} = \{(h_{t_1}^i, h_{t_2}^j)\}$, where $t_1 \neq t_2$. Moreover, $h_{t_1}^i$ and $h_{t_2}^j$ must be the same class of object and the distance between the two hypotheses must be less than a threshold $\phi_{threshold} = |t_1 - t_2| v_{max}$, where $1 \leq |t_1 - t_2| \leq t_{max}$. The term t_{max} is the size of the window frame; v_{max} represents the maximum speed of an object of a specific class.

The goal of the neighbourhood relations is to reduce the search space. Instead of searching over the entire solution space for the maximum posterior solution, the method only needs to search for trajectories within the neighbourhood graph of moving object hypotheses. Sibling edges indicate an exclusion relation between object hypotheses that are generated by the same moving object, so that if one is selected to form a track then the others are excluded. Besides, parent-child edges indicate a possible temporal association between hypotheses (possible data association).

The MCMC-based tracking method provides a useful framework to perform track management (Vu, 2009). Using the proposal distribution $q(\omega|\omega_{n-1})$ from Algorithm 3 and assuming, that at $n-1$ iteration, we have a sample $\omega_{n-1} = \{\tau_1, ..., \tau_K\}$ composed of K tracks formed by nodes of moving objects V, a new proposal w' and V^* are generated. V^* denotes the set of all unselected nodes in V that do not share any sibling edge with nodes in ω_{n-1}. The operations for track management provided by the aforementioned tracking method are: creation and deletion of tracks; split of a track and merging of two tracks; and the modification of specific joints inside an object trajectory promoted by information from other track or by the new track sample itself.

Once the object model was discovered using the DATMO solution based on MCMC, we build an evidence distribution for the class of objects based on the TBM. This distribution or mass assignment is done over the frame of discernment $\Omega = \{pedestrian, bike, car, truck\}$. We used two object models: box model and point model to establish the class evidence to each possible hypothesis. The basic belief assignment for lidar data m_l is defined as follows:

$$m_l(A) = \begin{cases} m_l(\{pedestrian\}) = \alpha_p & \text{if point model} = \text{true} \\ m_l(\Omega) = 1 - \alpha_p & \\ m_l(\{bike\}) = \alpha_b & \text{if box model} = \text{true } \textit{and} \text{ size } = \text{bike} \\ m_l(\Omega) = 1 - \alpha_b & \\ m_l(\{car\}) = \gamma\alpha_c & \text{if box model} = \text{true } \textit{and} \text{ size } = \text{car} \\ m_l(\{car, truck\}) = \gamma(1 - \alpha_c) & \\ m_l(\Omega) = 1 - \gamma & \\ m_l(\{truck\}) = \alpha_t & \text{if box model} = \text{true } \textit{and} \text{ size } = \text{truck} \\ m_l(\Omega) = 1 - \alpha_t & \end{cases} \tag{4.1.2}$$

where $A \in \Omega$. α_p, α_b, α_c, α_t are confidence factors obtained empirically from real data tests of the tracking process and represent how likely it is that the detected model and size represent a pedestrian, bike, car, or a truck respectively. When a pedestrian, a bike, or a truck is recognized, its model or size limit the uncertainty put in another class hypothesis.

However, due to the uncertainty from the α factors, it is possible that the model discovering or the size estimation information is not complete. For this reason a mass evidence is put in the ignorance hypothesis Ω. The *car* class is a special case we noticed from the experimentation and the results from Vu (2009). Due to the limited visibility of the lidar sensor, sometimes an object is considered a car even if the lidar frames up to current time have not sensed the edges of the box model for a car. Therefore, if a car sized object is detected, we do not only put evidence in the $\{car\}$ hypothesis but also on the $\{car, truck\}$. In order to do so we used a discounting factor γ which quantized the likelihood of the car $\{car\}$ being incomplete. After the discounting process is done the residual evidence is set to the Ω hypothesis.

4.1.2 Radar sensor

Radar data is comprised of a list of detected targets. These targets can be either static or dynamic (moving). However, when the targets are moving they are characterized by their relative moving speed. We use only the moving targets as the input for our multi-sensor fusion approach. Therefore, the input provided by radar sensor is composed of a list of moving targets represented by their pose and relative speed.

Multiple Object Tracking

Once we have the list of possible moving targets, it is imperative to track them and filter the real objects from the false positives. Moreover, filtering will estimate the state of moving objects, even if there are temporary occlusions or missing detections due to sensor noise.

In order to describe the multiple object tracking module we briefly describe the single object tracking and extend from it the general multiple object tracking solution we follow for radar data.

The moving object tracking problem for a single target can be mathematically formalized using a probabilistic representation. Therefore, it can be seen as a posterior probability using the recursive update equation of the Bayesian filter described in Section 2.5.1:

$$P(x_t|z_{1:t}) = \alpha P(z_t|z_{1:t-1}) \int P(x_t|x_{t-1}) P(x_{t-1}|z_{1:t-1}) dx_{t-1}, \tag{4.1.3}$$

where x_t represents the state vector of the moving object to be estimated and therefore tracked. The state vector usually contains kinematic data, including the position and velocity of the object. However, we can include other information such as the class of the object. $z_{1:t}$ is the set of measurements received up to time t.

The object state x_t is inferred only from the previous state x_{t-1} and object measurements $z_{1:t}$. The difference with SLAM formulation is that we do not have access to the control inputs of the object being tracked. There are three other factors to be considered to compute the state estimation: motion model of the moving object $P(x_t|x_{t-1})$; measurement model $P(z_t|x_t)$; and the posterior at time $t-1$ which finally becomes the prior at time t, $P(x_{t-1}|z_{1:t-1})$. The prior $P(x_0)$ describes the initial state of the tracked object represented by a stochastic process. Thus, $P(x_t|z_{1:t})$ is a probability distribution that describes our belief about the state of the tracked object at time t given the measurements z up to current time t.

The motion model $P(x_t|x_{t-1})$ describes how the state of the object evolves over

time. As there is no a priori information about the trajectory as well as the control inputs of the moving object. Usually, due to the lack of prior information about the moving object control inputs, a stochastic model is used to predict the possible motion of the object. Commonly, in intelligent vehicle applications, three main motion models are used: random, constant velocity and constant acceleration model. The simplest model is the random motion model which is often used for tracking pedestrians as they can change their velocity and direction of motion rapidly. In this model, pose of the object is predicted with an addition of zero-mean Gaussian noise whose variance grows with time. For vehicles, however, a more widely used motion model is the constant velocity model which estimates the position and velocity of the object using an acceleration noise to model the changes in velocity. In this model, the pose usually evolves linearly, which is often used when the precise dynamics of tracked objects are not known.

In practice, a moving object can change its dynamics over time (e.g., a moving vehicle stopping at a red light and moving again after the green light) and the motion models for each of these behaviours are different. This issue leads to the requirement of a more complete motion model definition method such as the Interacting Multiple Models (IMM) described in Section 2.3.4. IMM makes the assumption that, at a given time, the object moves according to one model from a predefined set of models. In this case, at each time step, we need to estimate the corresponding motion model in addition to the state of the object. We follow the IMM idea to describe the possible motion models of the tracked objects. Our implementation of IMM includes four motion models: stopped, constant acceleration, constant velocity, and turning.

Given the radar information for each target, we extract the information to form the object state. Thus, we represent the models of the objects parametrized by $S_x = \{x, y, \theta, c\}$ where x and are the position of the target, θ is its orientation, and c represents the class of the object which is inferred based on the estimation of the speed discovered over time.

Now, let us describe the multiple object tracking solution using the previous single object tracking formalization. We can directly extend the state vector to include all moving objects. However, as was covered in Section 2.3.3, there is one important additional problem to solve: data association between the measurements and objects, which aims to identify which measurement is generated by which object.

Let us consider the set of n moving objects $\{x_t^1, x_t^2, ..., x_t^n\}$ at time t. Thus, we can define a state vector X_t for all the objects at time t. The dynamic model for this state vector is $P(X_t|X_{t-1})$ and must define the joint evolution of all the objects. Usually, the

aforementioned problem is simplified by assuming independence between the objects motion. Thus, the joint distribution can be computed as:

$$P(X_t|X_{t-1}) = \prod_{i=1}^{n} P(x_t^i|x_{t-1}^i), \tag{4.1.4}$$

where $P(x_t^i|x_{t-1}^i)$ represents the motion model for object i.

In cluttered scenarios, it is difficult to identify which observation corresponds to which moving object. Therefore, a solution to the problem of Data Association becomes fundamental. As was covered in

In order to track moving objects, we need to sequentially estimate their state. The class information of the objects helps us in selecting the appropriate motion parameters for the state estimation and consequently selecting the appropriate motion model. Regarding the motion models, we use a classical IMM implementation composed of: stopped, constant velocity, constant acceleration, and turning motion models for each possible object (Kaempchen, 2004).

Our Data Association approach manages only data provided by radar, therefore we developed a common Multi Hypothesis Tracking approach. To control the growth of hypothesis tree we used a pruning technique based on a gating filter which uses the distances between observations and moving objects. Moreover, a N-Scan constraint was implemented to keep control of the growing hypothesis tree. We define the distance Dg as the gating factor and we only generate a hypothesis if the objects belong to the same class. Distance Dg was set empirically as 0.7m.

The object class is inferred from its estimated speed calculated after tracking processing. In order to assign a class to an object we propose the following class evidence distribution assignation for radar data m_r, also known as the basic belief assignment:

$$m_r(A) = \begin{cases} m_r(\{pedestrian, bike\}) = \alpha \\ m_r(\{pedestrian, bike, car, truck\}) = 1-\alpha \quad \text{if } object_{speed} < S_p \\ m_r(\{car, truck\}) = \beta \\ m_r(\{pedestrian, bike, car, truck\}) = 1-\beta \quad \text{if } object_{speed} > S_p \end{cases} \tag{4.1.5}$$

where $A \in \Omega$. α and β are the evidence factors that assign evidence to specific classes. S_p is a speed threshold used to discriminate slow objects from fast ones. If an object speed is greater than S_p it is most likely to be a car or a truck, else it can be of any class. However, the factor $1-\alpha$ assigns uncertainty to possible cases when it is most likely to be pedestrians moving slower than vehicles, e.g., rural roads or highways. $1-\beta$

represents the opposite case, if the object is moving faster than S_p, $1 - \beta$ represents the uncertainty from the estimation which can be caused by wrong associations.

4.1.3 Camera Sensor

Lidar and radar sensors provide useful information regarding the object's state. Basic information from the camera sensor does not contain any information regarding the object's pose. Although it is possible to perform 2D tracking using mono camera images we decided not do so in order to overcome future computational time constraints. We assume that information from lidar and radar could be enough to estimate kinetic information. Besides, we believe that including mono camera information only as an appearance-based source, allows us to show how the fusion approach can work with complementary data rather than only redundant measurements.

We use camera images to extract appearance information that leads to obtain an evidence distribution of the class of objects present in the image. We follow the image processing described in Section 3.5.3. Our camera sensor processing is divided into four steps. First, we build an object classifier using off-line training and several datasets. Second, we generate the object hypotheses where objects of interest can be present in the current image. After, we use the trained classifier to test the generated hypotheses and classify them according to the set of objects of interest (i.e., $\{pedestrian, bike, car, truck\}$). Finally, we build an evidence distribution for each classification based on the TBM framework (described in Section 2.5.1). The goal of camera processing is to deliver class evidence distribution for each possible object of interest. The first three steps of this processing were covered in Section 3.5.3. Therefore, we will proceed to detail the evidence distribution mass assignation using the TBM.

Evidence Distribution for Object Classification

The possible set of class hypotheses is set as $\Omega = \{pedestrian, bike, car, truck\}$. According to the evidential framework introduced in Section 2.5.1, Ω is known as the frame of discernment, 2^{Ω} being all the possible hypotheses for the evidence distribution of an object. Therefore, we need to establish the method to define the basic belief masses $m(A)$, for $A \in 2^{\Omega}$. This evidence assignment has to follow the restrictions stated in Equation 2.5.5.

Although we have built two main classifiers: one for pedestrians and one for vehicles (cars, trucks), we used them in cascade to form a single multiple class classifier. First, the pedestrian classifier is applied to recognize pedestrians in the image, if the

image is classified as no pedestrian then a vehicle classifier is applied. If after vehicle classifier, no object of interest is recognized, then we have no knowledge (ignorance) about the class of the object in the image.

The classifier needs a descriptor vector, extracted from an image, as input. If we use the entire image to look for objects of interest, the computational time and resources become so high that a real time classification is unreachable. As we have no clue where the objects of interest can be located, we use the output of the moving object detection processing from the lidar sensor. This output is a list of possible moving objects represented by a position and cluster of cells in the moving occupancy grid D. Using the calibration data and geometrical transformation we can translate the coordinates of the lidar objects into the camera image, we call these images patches: *regions of interest* (ROI). Therefore, the classifiers are applied over each region of interest instead of over the whole image. For each ROI, N_{sroi} overlapping sub regions are generated varying the scale of the original ROI.

As was mentioned in Section 3.5.3 we can extract classification confidence by counting the number of overlapping positives object classifications. The more overlapping areas of similar classifications, the greater the confidence level. Experimentally, we have noticed that false classifications have few overlapping areas. Figure 4.4 shows an example of the vehicle and pedestrian classifier outputs for each lidar ROI present in the scenario. The confidence of the classification c_c is delimited by the range $[0, 1]$, where 1 means N_{sroi} ROIs were classified as a particular object of interest and 0 means the opposite.

We have proposed different basic belief masses according to the class of object being detected. These basic belief assignment for camera sensor m_c can be summarized as follows:

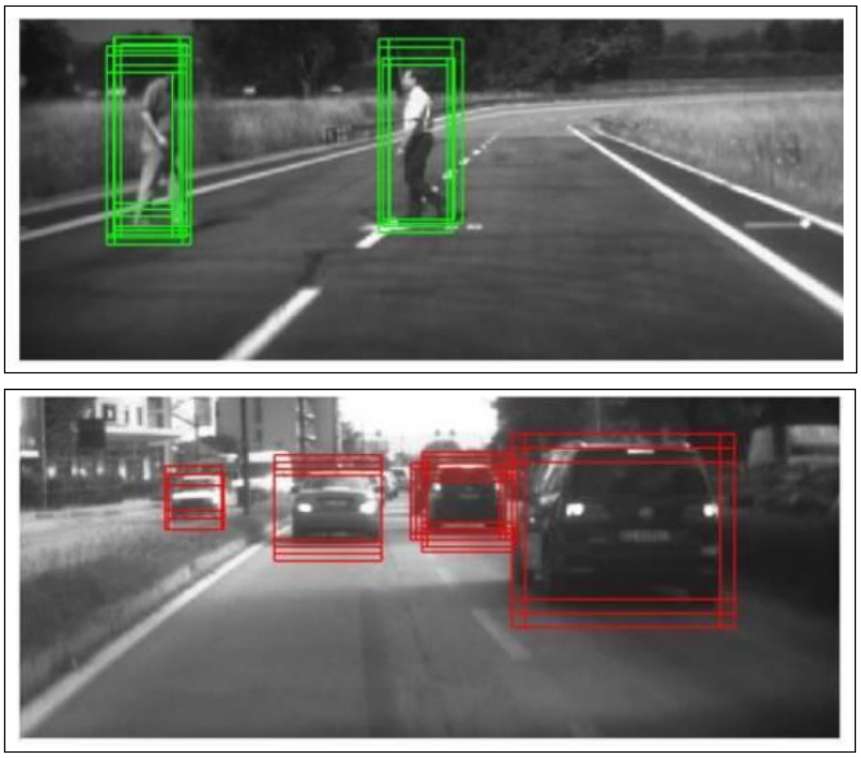


Figure 4.4: Output example from pedestrian (top) and vehicle classifiers (down) after being applied over the regions of interest from lidar moving object detection described in Section 4.1.1.

$$m_c(A) = \begin{cases} m_c(\{pedestrian\}) = \alpha_p c_c & \text{if } class = \text{pedestrian} \\ m_c(\{pedestrian, bike\}) = \alpha_p(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_p & \\ m_c(\{bike\}) = \alpha_b c_c & \text{if } class = \text{bike} \\ m_c(\{pedestrian, bike\}) = \alpha_b(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_b & \\ m_c(\{car\}) = \alpha_c c_c & \text{if } class = \text{car} \\ m_c(\{car, truck\}) = \alpha_c(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_c & \\ m_c(\{truck\}) = \alpha_t c_c & \text{if } class = \text{truck} \\ m_c(\{car, truck\}) = \alpha_t(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_t & \end{cases} \tag{4.1.6}$$

for $A \in \Omega$. α_p, α_c and α_t are discounting factors that represent how good is the classifier

to classify pedestrians, cars and trucks respectively. If an image is detected as pedestrian or bike, these results are still uncertain due to the nature of the classifier and the noise of the training data, therefore masses $1-(\alpha_p)$ and $1-(\alpha_b)$ are assigned to the ignorance hypotheses. Due to occlusions or to the inner class variation in the training dataset, a bike is classified as a pedestrian. These scenarios are represented by assigning a mass evidence to the hypothesis $\{pedestrian, bike\}$. If a car or a truck is detected, then it is still likely to be one of the two classes due to the similar datasets used to train these classifiers. Therefore, a mass evidence is first set to the hypothesis $\{car, truck\}$ and then the residual mass is set to the ignorance hypothesis to contemplate the uncertainty of the classification.

4.2 Multi-sensor fusion at tracking level

Figure 4.5 shows a schematic of the proposed fusion approach based on the classification information extracted from each of the three sensors. This schematic is located inside the fusion module of the architecture shown in Figure 5.2. The input of this method is composed of several lists of detected objects, their class information, the reliability of the sources of evidence, and the precision detection for certain type of classes. We build basic belief assignments for every object in the lists taking into account their class information. Using a proposed conjunctive rule of combination, we combine the classification information from classifier modules at a current time to obtain an instantaneous combination, later on the instantaneous class information is fused with previous combinations in a process we call dynamic combination. The final output of the proposed method comprises a list of objects with combined class information.

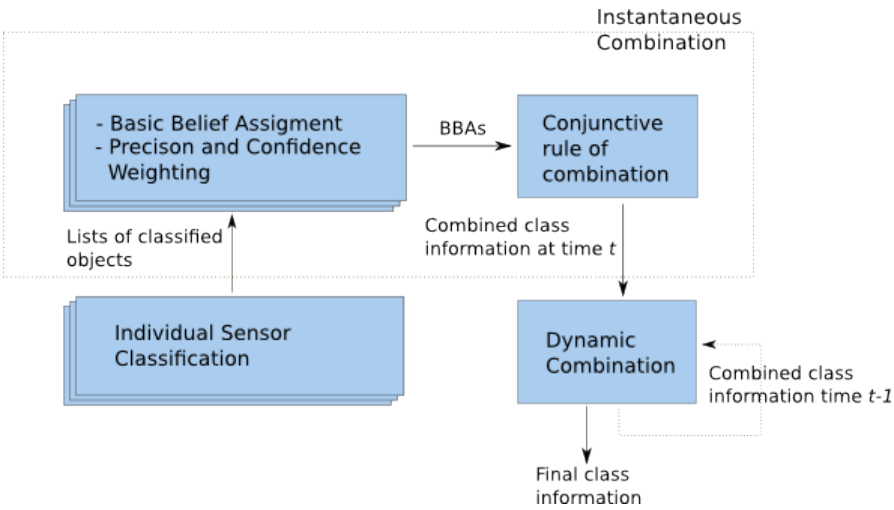


Figure 4.5: Schematic of the proposed fusion architecture.

Although tracking is performed individually for each sensor input (as shown by Figure 4.2), a global list of tracks $G_l(t)$ is maintained at each moment to store the current t global state of the tracked objects. When new data is available, this list becomes the previous state of the tracks $G_l(t-1)$ and is used by the individual tracking modules to help in the track-confirmation process of each local list i of tracks: $L_l^i(t)$. This process is part of the two-stage fusion architecture described in Figure 4.5, where the local lists are used to perform an instantaneous combination, and the global list is used to compute a dynamic combination. In the next sections we will describe the two stages of our fusion approach at tracking level and the final result obtained after the dynamic combination is computed. The inputs of this approach were described in previous sections.

4.2.1 Instantaneous Combination

According to Dempster-Shafer theory, let us define the frame of discernment Ω and the power set of Ω as the set of all possible class hypotheses for each source of evidence. Where Ω represents the set of all the classes we want to identify. Let's define m_b as the reference mass evidence and m_c as the mass evidence from the source we want to combine with. Finally, m_r represents the combined mass evidence.

In situations where the conflict mass is high, Dempster's combination rule generates counter-intuitive results, for this reason we have decided to adapt the combination rule proposed by Yager (1985) to obtain a more suitable rule of combination that avoids counter-intuitive results. The main idea is to move the conflict mass (K) to the set Ω. Which means transferring the conflict mass to the ignorance state instead of normalizing the rest of the masses. We do this expecting that future mass evidence will help to solve conflict states when two sources of evidence differently classify one object. The used rule of combination is stated as follows.

$$\begin{aligned} m_a(A) &= \sum_{B \cap C = A} m_b(B) m_c(C) \text{ for } A \neq \emptyset, \\ K &= \sum_{B \cap C = \emptyset} m_b(B) m_c(C), \\ m_a(\Omega) &= m'_a(\Omega) + K, \end{aligned} \tag{4.2.1}$$

where $m'_a(\Omega)$ is the BBA for the ignorance state and $m_a(\Omega)$ includes the added ignorance from the conflict states. This rule considers both sources of evidence as independent and reliable.

As we cannot assure the reliability of the evidence sources regarding the classification due to sensor limitations or miss classifications, we proposed to use a *discounting*

factor for each source of evidence (Smets, 2000). We believe that by doing this it will allow us to deal with reliability issues.

Let us define m_b as a reliable reference source of evidence and m_c as a relatively reliable source of evidence. We define $r_{bc} \in [0,1]$ as the reliability factor of m_c with respect to m_b. To make m_c reliable, we apply r_{bc} over the BBA of m_c. The evidence we take from the subsets of 2^Ω after applying the reliability factor should be considered ignorance, therefore is transferred to the set Ω:

$$\begin{aligned} m_c(A) =& r_{bc} \times m'_c(A), \\ m_c(\Omega) =& m'_c(\Omega) + \sum(1 - r_{bc} \times m'_c(A)) \\ & for A \subseteq 2^\Omega, A \neq \emptyset, A \neq \Omega, \end{aligned} \tag{4.2.2}$$

where m'_c represents the unreliable BBA. This equation means that we adjust the mass evidence of m_c according to how reliable it is compared with the reference source of evidence m_b. When m_c is as reliable as m_b ($r_{bc} = 1$), we get the original BBA for m'_c:

$$\begin{aligned} m_c(A) &= m'_c(A), \\ m_c(\Omega) &= m'_c(\Omega), \end{aligned} \tag{4.2.3}$$

There are scenarios where one of the sources of evidence is more precise than the other in identifying the class of a specific subset of the frame of discernment. We can include this imprecision using a similar approach to the one proposed above for the reliability description. However, our proposal focuses on specific subsets of the frame of discernment.

We consider $f_i \in [0,1]$ as the precision factor for the *ith* subset (hypothesis) of a particular belief function m_b. The greater the value the more precise the source evidence is about the mass assigned to the subset.

$$\begin{aligned} m_b(A_i) =& m'_b(A_i) \times f_i, \\ m_b(\Omega) =& m'_\Omega + \sum(1 - f_i) \times m'_b(A_i) \\ & for A_i \subseteq 2^\Omega, A_i \neq \emptyset, A_i \neq \Omega, \end{aligned} \tag{4.2.4}$$

here m'_b represents the reliable BBA. All the unallocated evidence will be placed in the Ω state because it is considered ignorance.

Once we have applied the reliability and precision factors, the combination rule in Equation 4.2.1 can be used. Several sources can be combined applying iteratively

the aforementioned rule of combination and using the fused evidence as the reliability reference source.

The final fused evidence contains the transferred evidence from the different sources. The criterion we used to determine the final hypothesis is based on the higher mass function value from the combined set, though it can be modified to be based on *belief* or *plausibility* degrees.

Track Association

Since we are performing the combination of different sources of evidence at current time t, we will call this *instantaneous fusion*. The inputs of this instantaneous fusion are the set of basic belief assignments described for the lidar, radar, and camera processing. However, as the lidar provides more reliable information due to the resolution of the sensor we consider m_l as the reference evidence distribution. Moreover, we do not need a data association process to relate the moving objects from lidar and camera because the camera uses ROI from lidar processing. Afterwards, the fused mass distribution is considered as the reference distribution and therefore combined with the radar mass assignment m_r. The association between lidar and radar objects is done using a gating approach between tracks based on the covariance matrices of the tracks from both sensors, this approach is based on the association techniques proposed by Bar-Shalom and Tse (1975) and Baig (2012). Also we include the idea of associating tracks that have parallel trajectories to perform track confirmation. The final combination represents the instantaneous combination.

In order to fuse the objects' states from two different tracks and to obtain a more detailed view of environment, we used a Bayesian fusion method to get combined objects' positions and therefore get better tracking results. The fusion of this information can be useful when there are many false alarms in the lists of tracks. Also, this allows to confirm tracks rapidly.

For the sake of clarity, let us describe the position of an object detected by lidar processing as $p_l = [d_l, \theta_l]^T$ and the position of an object detected by radar processing as $p_r = [d_r, \theta_r]^T$, where d_l and d_r represent the distance from the sensor to the detected object and θ_l and θ_r represent the angle to the detected objects. If we assume that $x = [d, \theta]^T$ is the true position of the detected object, then we can infer the probability that lidar and radar processing detect the object as follows:

$$
\begin{aligned}
P(p_l|x) =& \frac{exp-(d_l - x)^T R_l^{-1}(p_l - x)}{2\pi\sqrt{|R_l|}}, \\
P(p_r|x) =& \frac{exp-(d_r - x)^T R_r^{-1}(p_r - x)}{2\pi\sqrt{|R_r|}},
\end{aligned}
\tag{4.2.5}
$$

where R_l and R_r are covariance matrices that represent the uncertainty from the range and angle for lidar and radar processing, respectively. These matrices are extracted from data provided by the sensor vendors.

Bayesian fusion is a state-of-the-art approach to combine information. We use the definitions from Section 2.5.1 to represent the fused position probability of an objectś position as follows:

$$
P(p|x) = \frac{exp-(p - x)^T R^{-1}(p - x)}{2\pi\sqrt{|R|}}, \tag{4.2.6}
$$

where the fuse position p and the covariance matrix R are defined as:

$$
\begin{aligned}
p =& \frac{p_l/R_l + p_r/R_r}{1/R_l + 1/R_r}, \\
R =& \frac{1}{1/R_l + 1/R_r},
\end{aligned}
\tag{4.2.7}
$$

where the fused position represents the kinetic part of our object representation. The evidence distribution, provided by the evidential fusion of class information, complements the aforementioned part to build the composite representation.

4.2.2 Dynamic Combination

It is evident that including information from the previous combination can add valuable prior evidence to the current available evidence. For example, the evidence about the class of the tracked objects stored in previous combinations can be updated with the new sensor measurements at current time. The TBM mechanisms allow to transfer the new mass evidence from the sensor measurements to only the hypotheses where an updating process needs to be done. Regarding this topic, and taking advantage of the proposed general framework architecture, we introduce Equation 4.2.8 as an extension of the proposed instantaneous fusion to include mass evidence from previous combinations (i.e., time $t-1$):

$$
m_r^t(A) = m_r(A) \oplus m_r^{t-1}(A), \tag{4.2.8}
$$

where $m_r(A)$ represents the instantaneous fusion at time t. The operator $\oplus$ follows the same combination rule defined in Equation 4.2.1, which is also used to obtain the instantaneous fusion. Following this extension we can notice that the combined mass of the list objects from all the previous times is represented by $m_r^{t-1}(A)$.

Hence, the final output of the DATMO component and whole perception solution is composed not only of the list of moving objects' tracks described by object state, but each object class is also described by a fused evidence distribution over the frame of discernment Ω. An object class decision can be obtained by taking the hypothesis with the maximum mass or by an interpretation within the TBM, such as degree of belief or plausibility.

4.3 Experimental Results

Experiments were conducted using four datasets obtained from the sensor set-up described in Section 3.3. We tested the multi-sensor fusion approach at the tracking level in two main scenarios: urban and highway. The objective of these experiments was to verify if the results from our proposed approach improves the preliminary classification results provided by the individual object classifier modules. Our implementation of the lidar based classifier is an adaptation of the method proposed by Vu (2009). Our implementation of the radar based classifier can be considered as a state-of-the-art radar approach.

We followed the previously set frame of discernment $\Omega = \{pedestrian, bike, car, truck\}$ and therefore the set of all possible 2^{Ω} classification hypotheses for each source of evidence: lidar, radar, and camera. For display purposes, we used the hypothesis with the maximum mass as the final class of the tracked object.

Lidar processing is able to provide a classification for all possible kinds of objects using cell clusters and the model-based approach. It has good performance when identifying cars and trucks, but a poor performance when it comes to pedestrians or bikes. We represent this behaviour by setting high confidence factors for car and truck classifications and low confidence factors for pedestrian and bike classifications in Equation 4.1.2. However, the aforementioned confidence is corrected over time when more frames are available and the model-based approach estimates a better model for each tracked object.

Figures 4.6 and 4.7 show the results of the fusion approach as part of a whole SLAM+DATMO solution. We tested our proposed approach in several urban and high-

Table 4.1: Number of car/truck miss-classifications from individual lidar and radar tracking modules and from the proposed fusion approach.

Dataset	Number of objects	Lidar/image classifier	Radar tracker/classifier	Fusion approach
Highway 1	110	12	19	6
		10.9%	17.2%	5.4%
Highway 2	154	17	23	7
		11%	14.9%	4.5%
Urban 1	195	29	36	20
		14.8%	18.4%	10.2%
Urban 2	233	39	48	24
		16.7%	20.6%	10.3%

way scenarios. We obtained good results in both scenarios compared with the individual classification inputs.

Figure 4.6 (a) shows how the proposed approach identifies the class of the two moving objects present in the scene: a car and a bike. On the contrary, in Figure 4.7 (b) one pedestrian is missing because none of the classification sources provided evidence to support the *pedestrian* class. This is due to the noise present in both the lidar and radar measurements. However, in future frames the detection and classification of the missing pedestrian becomes clear.

The car in Figure 4.7 (b) and the truck in Figure 4.6 (a) are not yet classified by the lidar processing because they have just appeared few frames before in the field of view. However, fusing the evidence put in the ignorance hypothesis with the classification from the camera classifier, the proposed approach can correctly identify both vehicles at early time. Mass evidence supporting these two classification hypotheses becomes higher in posterior frames when lidar processing provides more evidence about the correct class of objects.

Figure 4.7 (b) shows how, despite the lack of texture in the image to identify the two vehicles in the front, evidence from the lidar processing helps to correctly identify them. This is a common scenario when weather conditions make the correct camera-based classification difficult.

Tables 4.1 and 4.2 show quantitative results obtained by the proposed fusion approach. Also, these tables show a comparison between the results obtained by the proposed fusion approach and the individual moving object trackers/classifiers regard-

Table 4.2: Number of pedestrian/bikes miss-classifications from individual lidar and radar tracking modules and from the proposed fusion approach.. Highway datasets do not contain any pedestrian or bike.

Dataset	Number of objects	Lidar/image classifier	Radar tracker/classifier	Fusion approach
Urban 1	52	23	24	11
		44.2%	46.1%	21.1%
Urban 2	58	26	28	14
		44.8%	48.2%	24.1%

ing the total number of object miss-classifications per dataset. We used four datasets provided by the vehicle demonstrator described in Section 3.5, two of them from highway scenarios and two from urban scenarios. We can see that the proposed fusion approach reduces the number of miss-classifications in all the datasets. From the highway datasets, we can see that the individual trackers and classifiers perform better than in urban scenarios. However, the degree of improvement achieved by our proposed fusion approach at tracking level is high even in the highway scenarios. This improvement is due to the ability our fusion approach to discard ghost objects with high lack of evidence about their class.

Due to the specific nature of the pedestrian and vehicle classifiers, we divided the experiments to separately analyse the improvement in pedestrian/bike and car/truck classifications. One can see how different the individual performances are of the evidence sources to detect specific class objects, this is highly related to the nature of the sensor information they use to provide class hypotheses. The proposed fusion approach combines the class evidence provided by each source of evidence among its reliability and precision factors to obtain a better classification of the moving objects.

Table 4.2 shows that our lidar-based and radar-based tracker do not perform well when detecting pedestrians and bikes. However, when we fuse position information from both lidar and radar measurements plus the classification from camera, the improvement becomes relevant (a reduction of about the half of mis-classifications) and only few pedestrians and bikes are mis-classified. In the case of the lidar-based tracker, small clusters of cells cannot be identified as pedestrians because no appearance information can discriminate them from other small-sized objects, e.g. poles and traffic signs.

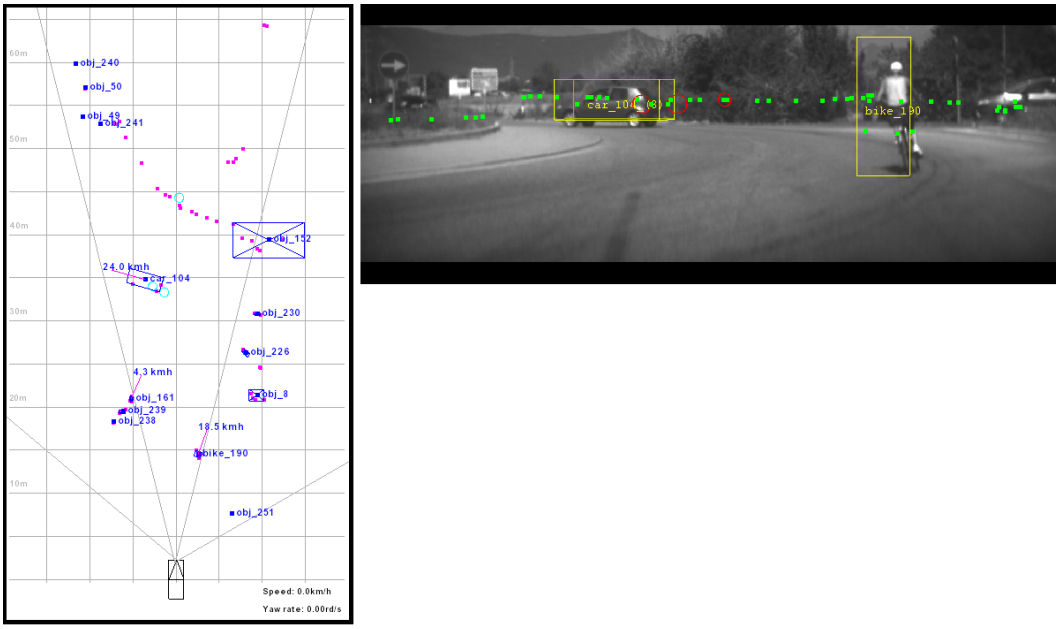
Although our fusion approach at tracking level works in late levels of the DATMO

solution, it does not merely consider the classification information as an aggregate to the final result. The evidential classification distribution for each tracked object enriches the object state representation and provides a non-definitive class decision to the next components of the perception solution, i.e. the decision-making task.

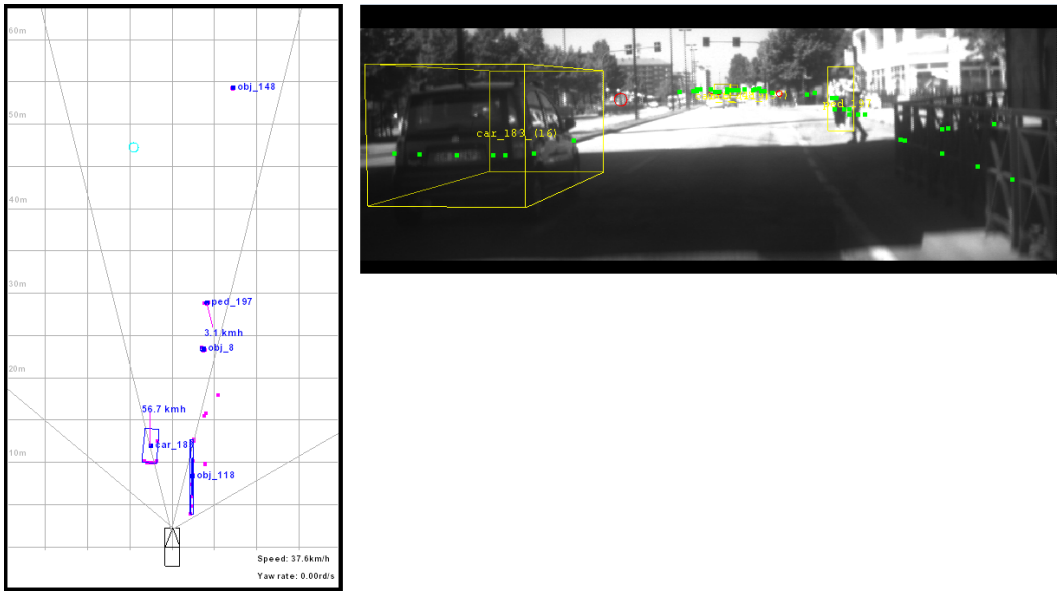
4.4 Summary

In this chapter we have presented our multi-sensor fusion approach at tracking level. We have described the general architecture of the fusion approach and detailed every component of the architecture. The sensor measurements have been processed to generate a map of the environment and detect the moving objects present in it. Two tracking modules were constructed, one from lidar scans and one from radar targets. Classification information is extracted from the three sensors and is used to generate evidential mass distributions that are the inputs for the classification fusion approach. Due to the better precision of lidar measurements we consider the lidar tracker as a reference for the fusion process. An evidential operator powers our fusion approach. First, the fusion approach performs an instantaneous fusion between the classification distributions at current time t. Second, the result of the instantaneous fusion is combined with the previous fusion, at time $t-1$ to update the evidential classification distribution.

Several experiments were performed in two different scenarios: highways and urban areas. The results showed considerable improvements compared to the results from the single-sensor trackers and classifiers. The results of our fusion approach motivated the proposal of the fusion approach at detection level that will be presented in the next chapter. The idea is to analyse if including classification information at early levels of the DATMO component can improve more the final result of the perception task.

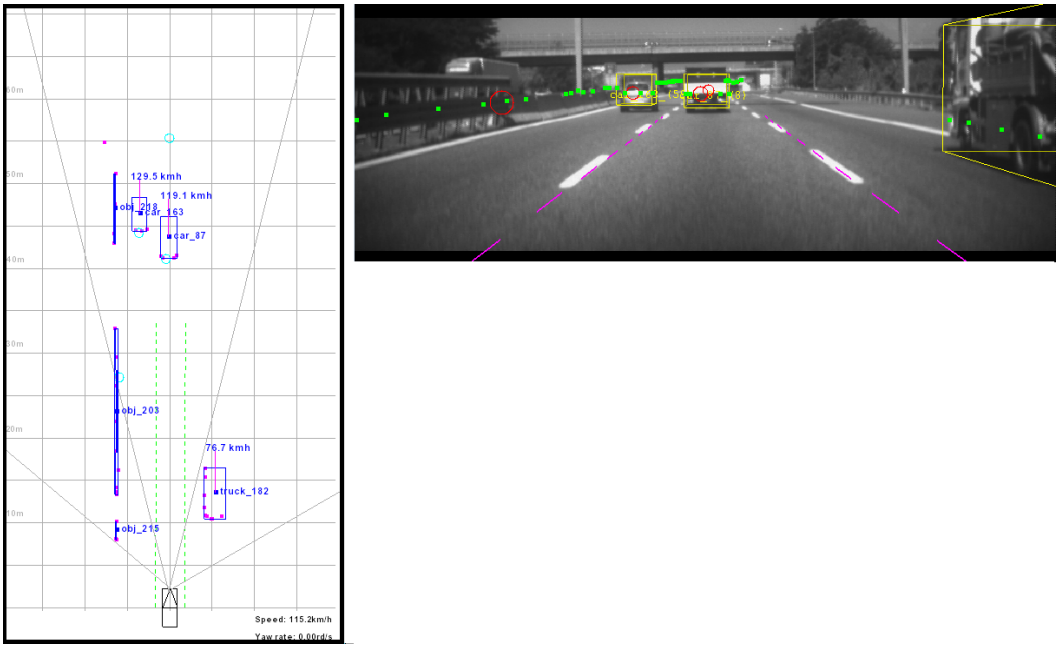


(a)

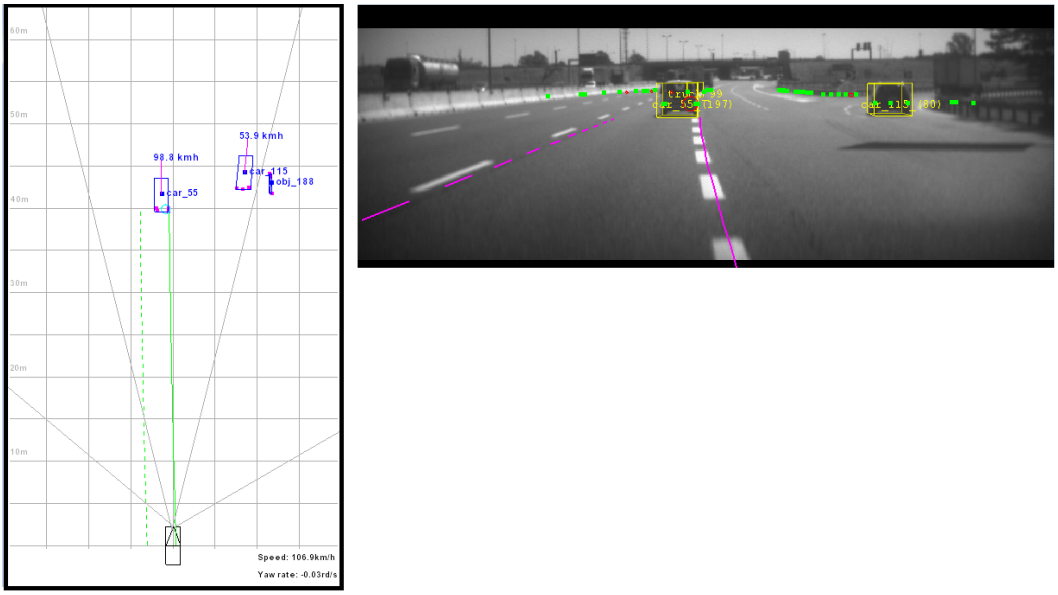


(b)

Figure 4.6: Results from the proposed fusion approach for moving object classification in urban scenarios (a,b) . The left side of each image represents the top view representation of the scene (static and moving objects) showed in the right-side image. Bounding shapes and class tags in the right side of each image represent classified moving objects.



(a)



(b)

Figure 4.7: Results from the proposed fusion approach for moving object classification in (a,b) highway scenarios. The left side of each image represents the top view representation of the scene (static and moving objects) showed in the right-side image. Bounding shapes and class tags in the right side of each image represent classified moving objects.

CHAPTER 5

Multi-sensor fusion at detection level

FOLLOWING the analysis from the previous chapter, in this chapter we present an evidential fusion framework to represent and combine evidence from multiple lists of sensor detections. This fusion framework considers the position, shape, and class information to associate and combine sensor detections, prior performing the tracking process. The fusion approach includes the management of uncertainty from detection and integrates an evidence distribution for object classes as a key part of a composite object representation. Although the proposed method takes place at the detection level inside DATMO, we propose a general architecture to include it as a part of a whole perception solution.

The tracking process assumes that its inputs correspond uniquely to moving objects, and it then focus on data association and tracking problems. However, in most of the real outdoor environment scenarios, tracking inputs include non-moving detections, such as noisy measurements or static objects. Technical limitations of sensors contribute in some degree to these imprecisions. For example, radar measurements have ground noise (climatic perturbations, floor of the sea), video images have non-stable backgrounds (trees on the wind, changing light conditions, moving camera), lidar data is affected for non-reflective surfaces, and may include non-moving targets or spurious ground measures. Detecting correctly moving objects is a critical aspect of a moving object tracking system; therefore, many sensors are usually part of common intelligent vehicle systems to overcome individual sensor limitations.

Knowing the class information when the detection step is performed could iteratively improve the final result of the DATMO task. Hereby, we propose a DATMO architecture where class information is included to enrich the knowledge about the

possible objects of interest around the vehicle. We propose an evidence updating algorithm to keep the kinetic and appearance information of the detected objects up-to-date. This updating algorithm is based on the TBM. Moreover, the composite object representation is kept through the entire tracking process, and hence it is updated over time when new data is available. Figure 5.1 shows the generic architecture of the proposed fusion approach which includes classification information at the object detection level to combine object detections from different sensors.

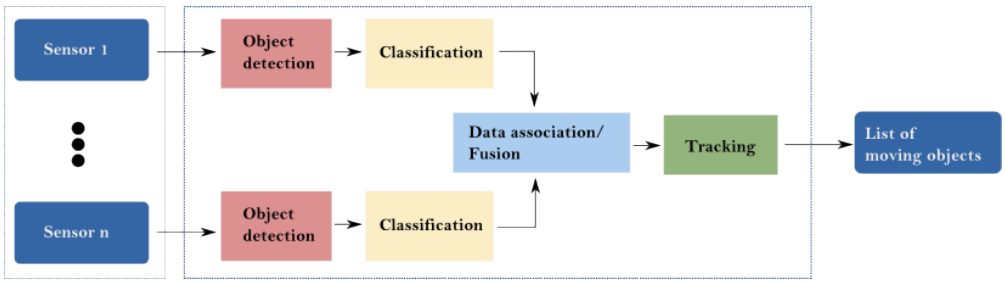


Figure 5.1: General architecture of the fusion approach at detection level.

Figure 5.1 shows the generic architecture of our DATMO solution for many sensor inputs. However, in order to describe our method we will focus on three main sensors: lidar, radar, and camera. Figure 5.2 shows the schematic of our proposed method taking into account three main sensor inputs. Also, it shows the interaction between the detections and classification modules that will be described in detail in the following sections of this chapter.

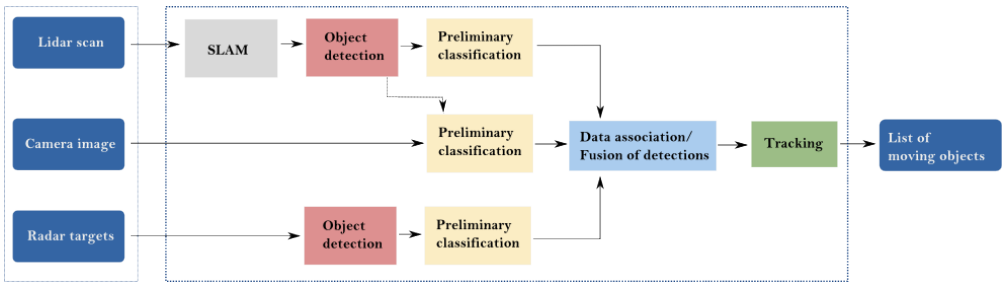


Figure 5.2: General architecture of the fusion approach at detection level for three main sensors: lidar, radar, and camera.

Classification of moving objects is needed to determine the possible behaviour of the objects surrounding the vehicle, and it is usually performed at tracking level. Knowledge about the class of moving objects at detection level can help to improve not only the detection rate, but also the object tracking process, reason about their behaviour, and decide what to do according to their nature. Moreover, object classification is a key factor that enables a better understanding of driving situations which are considered in the final stage of an intelligent application to perform specific actions.

As we did in Chapter 4, before describing the multi-sensor fusion approach, first, we define the inputs and data processing modules that are needed to fuse the object information at detection level. We present our moving object detection and classification modules for each of the three sensors involved in the architecture. Later on, we describe a proposed data association method for moving object detections. Afterwards, we introduce the fusion module using the object representations created by the first modules and using the relations found by the data association method. Finally, we present a tracking process that uses the fused representations and constantly updates the object's state and class information. Several experiments are performed to test and evaluate this fusion approach. Moreover, a comparison with the fusion approach at tracking level is done in order to review the degree of improvement obtained when classification information is included in the early stages of the DATMO component. Finally, a summary of the described topics is presented at the end of the chapter.

5.1 Moving object detection and classification

We follow the same sensor architecture detailed in Section 3.3 to describe and develop our fusion approach at detection level. Three main sensors are taken as inputs for our proposed solution. These three sensors are lidar, radar, and camera. Figure 3.7 shows the sensor set-up within our proposed DATMO solution.

Usually, object detections are represented by their position and shape features. We believe that class information can be important to consider within fusion at the detection level. This information can help to identify associations between lists of detections provided by sensors. However, at detection level, there is not enough certainty about the class of the object. Hence, keeping only one possible class hypothesis per detection disables the possibility of rectifying a premature decision.

Our detection representation is compose of two parts. The first part includes position and shape information in a two dimensional space. The second part includes an evidence distribution $m(2^{\Omega})$ for all possible class hypotheses, where $\Omega = \{pedestrian, bike, car, truck\}$ is the frame of discernment representing the classes of moving objects of interest.

In the following subsections we describe how we construct the object description for each sensor input. This description includes a kinetic part and an appearance part defined by an evidence distribution of class hypotheses. This object description is then used by the fusion approach to deliver a fused list of object detections that will be used to perform tracking.

5.1.1 Lidar sensor

Lidar sensor plays an important role in our proposed solution. It is used to perform the SLAM task in order to detect moving objects around the vehicle. We follow the same SLAM approach described in Section 4.1.1 but only until the moving object detection stage. This means that no tracking is performed at this level. The tracking process takes place after our fusion approach at detection level.

Raw lidar scans are processed to build a static map and to detect moving objects following the probabilistic grid described in Section 3.5.1. We incrementally integrate new lidar scans into a local 2D probabilistic occupancy grid. Inconsistencies through time in this occupancy grid allow the method to detect moving parts of the environment. The points observed in free space are classified as dynamic whereas the rest are classified as static. Moving cells are placed in an occupancy grid D for dynamic objects. Using the clustering process we identify groups of points inside D that could describe moving objects. The list of possible moving objects is taken as a preliminary list provided by lidar sensor processing.

The first part of the object representation can be obtained by analysing the shape of the detected moving objects. In the case of large object detections this description is modelled by a box $\{x, y, w, l, c\}$, where x and y are the center of the box, w and l are the width and length according to the class of object c. For small object detections (mainly pedestrians) a point model $\{x, y, c\}$ is used, where x, y and c represent the object center and class of the object, respectively. The position and size of the object is obtained by measuring the detected objects in the 2D occupancy grid. The class of the object is inferred from the visible size of the object and follows the same fixed fitting-model approach described in the Section 4.1.1. However, no precise classification decision can be made due to the temporary visibility of the moving objects. For example, if one object is detected and the width of its bounding box is less than a threshold ω_{small} we may think the object is a pedestrian or a bike but we can not be sure of the real size of the object. On the contrary, if the width of the object is greater than ω_{small} the object is less likely to be a pedestrian or bike, but rather a car or a truck.

We assume the class *car* as a fixed rectangular box of 1.7m in width by 4.5m in length. A *truck* is assumed to be a rectangular box with a width of 2.5m and length of between 9m and 12m. A bike is represented as a box with a length of 2.1m and a width of 0.5m. A pedestrian is assumed to be a box of 0.5m by 0.5m. We use a best fitting approach to preliminary decide the most probable class of the moving object. However, instead of keeping only one class decision, we define a basic belief assignment m_l which

describes an evidence distribution for the class of the moving object detected by lidar. $m_l(A)$ for each $A \in \Omega$ is defined as follows:

$$m_l(A) = \begin{cases} m_l(\{pedestrian\}) = \alpha_p & \text{if class} = \text{pedestrian} \\ m_l(\Omega) = 1 - \alpha_p & \\ m_l(\{bike\}) = \gamma_b \alpha_b & \text{if class} = \text{bike} \\ m_l(\{bike, car, truck\}) = \gamma_b(1 - \alpha_b) & \\ m_l(\Omega) = 1 - \gamma_b & \\ m_l(\{car\}) = \gamma_c \alpha_c & \text{if class} = \text{car} \\ m_l(\{car, truck\}) = \gamma_c(1 - \alpha_c) & \\ m_l(\Omega) = 1 - \gamma_c & \\ m_l(\{truck\}) = \alpha_t & \text{if class} = \text{truck} \\ m_l(\Omega) = 1 - \alpha_t & \end{cases} \tag{5.1.1}$$

where $\alpha_p, \alpha_b, \alpha_c$ and α_t represent evidence of the class and are fixed according to the performance of the laser processing. Also, these factors represent the lidar's ability to detect pedestrians, bikes, cars and trucks, respectively. γ_b and γ_c are discounting factors that indicate the uncertainty of the lidar processing for mis-detecting a bike, a car or a truck. If a *pedestrian* is detected, $1 - \alpha_p$ indicates the uncertainty of being correct in the detection.

When a bike is detected, due to visibility issues the detected object can still be a part of a *car* or a *truck*, for that reason evidence is also put in $\{bike, car, truck\}$. However, for the same *bike* example detection, due to the visible size of the object, we are almost sure this object is not a pedestrian. For the same reason, when a *truck* is detected, we are almost sure it cannot be a smaller object, for that reason no evidence is put in another class hypothesis. In all the cases, the ignorance hypothesis Ω represents the lack of knowledge and the general uncertainty on the class hypotheses.

5.1.2 Radar sensor

Radar targets are considered as preliminary moving object detections. To obtain the object description for each target we follow the same methodology described in Section 4.1.2, but rather than performing tracking we only focus on the moving object description. This means that we no longer use the estimate target speed to determine the preliminary class of the objects. Instead, we use the relative target speed deliver

by the sensor. The kinetic and class presentations are constructed in the same way. Therefore, we use the basic belief assignment m_r proposed in Equation 4.1.5

$$m_r(A) = \begin{cases} m_r(\{pedestrian, bike, car, truck\}) = \alpha & \text{if } object_{speed} < S_p \\ m_r(\{pedestrian, bike\}) = 1 - \alpha & \\ m_r(\{pedestrian, bike, car, truck\}) = 1 - \beta & \text{if } object_{speed} > S_p \\ m_r(\{car, truck\}) = \beta & \end{cases} \tag{5.1.2}$$

5.1.3 Camera sensor

We use camera images due to the high appearance information inside them. Information from camera images leads us to obtain another evidence distribution of the class of objects present in the scenario. We follow the image processing described in Section 3.5.3. This means that for hypotheses generation, we first build a S-HOG descriptor for each section of the image we want to classify. For hypothesis verification, we use the object classifiers we built to classify pedestrians, cars and trucks. In order to reduce the search space for hypotheses generation inside the camera image, we use Regions of Interest (ROIs) provided by the lidar processing. As was mentioned in Section 3.5.3, we can extract classification confidence by counting the number of overlapping positives object classifications. Hence, as was described in Section 4.1.3, we obtain a classification confidence based on the number of overlapping areas with the same classification tag.

Although the camera processing method uses ROIs to perform the classification process, it also generates several sub regions inside each ROI in order to cover many possible scale and size configurations. Sometimes a ROI can contain more than one object of interest. A common example occurs when a group of pedestrians is moving very close and the lidar processing detects them as a only one object. In this case, if the classifiers detect more than one object of interest inside the ROI, the single object is separated and the size (the width of the ROI) and class of object are updated. This means that although lidar processing assumes there is one object, camera processing considers it as many. Later on, the fusion process takes this into account to combine the available information and perform the object detection fusion.

Once we have obtained the object classification for each ROI, we generate a basic belief assignment m_c following the Equation 4.1.6. This belief assignment represents the evidence distribution for the classes hypotheses in Ω of each object detected for camera processing.

$$m_c(A) = \begin{cases} m_c(\{pedestrian\}) = \alpha_p c_c & \text{if } class = \text{pedestrian} \\ m_c(\{pedestrian, bike\}) = \alpha_p(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_p & \\ m_c(\{bike\}) = \alpha_b c_c & \text{if } class = \text{bike} \\ m_c(\{pedestrian, bike\}) = \alpha_b(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_b & \\ m_c(\{car\}) = \alpha_c c_c & \text{if } class = \text{car} \\ m_c(\{car, truck\}) = \alpha_c(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_c & \\ m_c(\{truck\}) = \alpha_t c_c & \text{if } class = \text{truck} \\ m_c(\{car, truck\}) = \alpha_t(1 - c_c) & \\ m_c(\Omega) = 1 - \alpha_t & \end{cases} \tag{5.1.3}$$

5.2 Fusion at detection level

Once we have described the moving object detection modules for each sensor and defined a composite object representation, the next task to describe is the fusion of object detections. Hence, we propose a multi-sensor fusion framework placed at the detection level. Although this approach is presented to work with three main sensors, it can be extended to work with more sources of evidence by defining extra detection modules that are able to deliver the object representation previously defined.

Figure 4.5 shows the general architecture of the proposed fusion approach. The inputs of this method are several lists of detected objects and their class information represented as basic belief assignments (BBAs). This means, each object detection is represented by its position, size and an evidence distribution of class hypotheses. The reliability of the sources of evidence is encoded in the BBAs by applying the confidence and discounting factors. Class information is obtained from the shape, relative speed and visual appearance of the detections. The final output of the fusion method comprises a fused list of object detections that will be used for the tracking module to estimate the moving object states and deliver the final output of our DATMO solution.

5.2.1 Data association

When working with many sources of evidence as in the object detection level, it is important to consider the problem of data association before performing the fusion of information. This means, we need to find which object detections are related from the different lists of detections (sources of evidence) in order to combine them. In this section, we describe our data association method as the first step of our fusion approach. Afterwards, we introduce the fusion approach once we have found the object detection associations. Later on, we describe how the combined list of detected objects is used in the tracking of moving objects.

The combination of information from different sources of evidence at the detection level has the advantage of increasing the reliability of the detection result by reducing the influence of inaccurate, uncertain, incomplete, or conflicting information from sensor measurements or object classification modules. One way to obtain this is having redundant or complementary information. Redundancy is typically achieved by overlapping the sensors' field of views. Complementariness can come from the diverse nature of the information provided by the sensors.

Let us consider two sources of evidence S_1 and S_2. Each of these sources provides a list of detections denoted by $A = \{a_1, a_2, .., a_b\}$ and $B = \{b_1, b_2, ..b_n\}$ respectively. As was mentioned before, in order to combine the information of these sources we need to find the associations between the detections in A and B. In other words, to find which detections belong to the same object. All possible associations can be expressed as a matrix of magnitude $|A \times B|$ where each cell represents the evidence m_{a_i,b_j} about the association of the elements a_i and b_j for $i < |A|$ and $j < |B|$.

We can define three propositions regarding the association of the detections in A and B:

- $P(a_i, b_j) = 1$: if a_i and b_j belong to the same object
- $P(a_i, b_j) = 0$: if a_i and b_j do not belong to the same object
- $P(a_i, b_j) = \Omega$: ignorance about the association of the detections a_i and b_j

Now, let us define $\Omega_d = \{1, 0\}$ as the frame of discernment of each m_{a_i,b_j}, where $\{1\}$ means that detections a_i and b_j belong to the same object, and $\{0\}$ otherwise. Therefore, $m_{a_i,b_j}(\{1\})$ and $m_{a_i,b_j}(\{0\})$ quantify the evidence supporting the proposition $P(a_i, b_j) = 1$ and $P(a_i, b_j) = 0$ respectively, and $m_{a_i,b_j}(\{1, 0\})$ stands for the ignorance, i.e., the evidence that can not support the other propositions.

The three different propositions or hypotheses can be addressed by representing the similarity of the detections in A and B, i.e., m_{a_i,b_j} can be defined based on similarity measures between detections a_i and b_j.

Sensors S_1 and S_2 can provide detections of a different kind. These detections can be represented by a position, shape, or appearance information, such as class. Hence, m_{a_i,b_j} has to be able to encode all the available similarity information.

Let us define m_{a_i,b_j} in terms of its similarity value as follows:

$$\begin{aligned} m_{a_i,b_j}(\{1\}) =& \alpha_{i,j}, \\ m_{a_i,b_j}(\{0\}) =& \beta_{i,j}, \\ m_{a_i,b_j}(\{1,0\}) =& 1 - \alpha_{i,j} - \beta_{i,j}, \end{aligned} \tag{5.2.1}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ quantify the evidence supporting the singletons in Ω_d for the detections a_i and b_j, i.e., the similarity measures between them.

We can define m_{a_i,b_j} as the fusion of all possible similarity measures to associate detections a_i and b_j. Therefore, we can assume that individual masses of evidence carry specific information about these two detections. Let us define m^p as the evidence measures about the position similarity between detections in A and B provided by sources S_1 and S_2 respectively; and m^c as the evidence measures about the appearance similarity.

Following the analysis made in Section 2.5.1, we use Yagers's combination rule defined in Equation 2.5.12 to represent m_{a_i,b_j} in terms of $m^p_{a_i,b_j}$ and $m^c_{a_i,b_j}$ as follows:

$$\begin{aligned} m_{a_i,b_j}(A) =& \sum_{B \cap C = A} m^p_{a_i,b_j}(B) m^c_{a_i,b_j}(C), \\ K_{a_i,b_j} =& \sum_{B \cap C = \emptyset} m^p_{a_i,b_j}(B) m^c_{a_i,b_j}(C), \\ m_{a_i,b_j}(\{\Omega_d\}) =& m'_{a_i,b_j}(\{\Omega_d\}) + K_{a_i,b_j}, \end{aligned} \tag{5.2.2}$$

where $m^p_{a_i,b_j}$ and $m^c_{a_i,b_j}$ represent the evidence about the similarity between detections a_i and b_j taking into account the position information and the class information, respectively.

Once the matrix $M_{A,B}$ is built, we can analyse the evidence distribution m_{a_i,b_j} for each cell to decide if there is an association ($m_{a_i,b_j}(1)$), it there is not($m_{a_i,b_j}(0)$), or if we have not enough evidence to decide ($m_{a_i,b_j}(1,0)$) which can probably be due to noisy detections. In the next subsections we will describe how to calculate these fused evidence distributions using similarity evidence from detections.

When two object detections are associated, the method combines the object representations by fusing the evidence distributions for class information. This fusion is achieved by applying the combination rule described in Equation 2.5.12. The fused object representation composed of the kinetic (position) and appearance (shape and class information) part is passed as input to the tracking stage to be considered in the motion model estimation of the moving objects. Non-associated object detections are passed as well expecting to be deleted by the tracking process if they are false detections or to be verified as real objects in case there is future information that corroborates these detections as real.

It is important to notice that not all the sensors provide the same amount and type of information. For example, radar data do not include information about the shape of the target, while lidar data can provide information about the position and the shape of the object. If two associated detections have complementary information, this is pass directly to the fused object representation; if the information is redundant, it is combined according to its type. For the position, we use the Bayesian fusion presented in Equation 4.2.7. Shape information is usually provided only by the lidar. As stated above, class information is combined using the evidential combination rule from Equation 2.5.12.

In the next section we review our proposed methods to extract similarity information from the position and class of the detections. This information is included in Equation 5.2.2 to decide if two detections are associated.

Position similarity

According to the position of two detections a_i and b_j, we encode their similarity evidence in $m^p_{a_i,b_j}$. Based on their positions, we can define function d_{a_i,b_j} as a distance function that satisfies the properties of a pseudo-distance metric. We choose Mahalanobis distance due to its ability to include the correlations of the set of distances (Bellet *et al.*, 2013). Therefore, a small value of d_{a_i,b_j} indicates that detections a_i and b_j are part of the same object; and a large value indicates the opposite. All the propositions for $m^p_{a_i,b_j}$ belong to the frame of discernment Ω_d . Hence, the BBA for $m^p_{a_i,b_j}$ is described as follows:

$$\begin{aligned} m^p_{a_i,b_j}(\{1\}) &= \alpha f(d_{a_i,b_j}), \\ m^p_{a_i,b_j}(\{0\}) &= \alpha(1 - f(d_{a_i,b_j})), \\ m^p_{a_i,b_j}(\{1,0\}) &= 1 - \alpha, \end{aligned} \tag{5.2.3}$$

where $\alpha \in [0,1]$ is an evidence discounting factor and $f(d_{a_i,b_j}) \rightarrow [0,1]$. The smaller the distance, the larger value given by function f. In our case we choose f as:

$$f(d_{a_i,b_j}) = exp(-\lambda d_{a_i,b_j}), \tag{5.2.4}$$

where λ is used as a threshold factor that indicates the border between close and far distances.

Class dissimilarity

Contrary to the evidence provided by position, class information does not give direct evidence that supports the proposition $P(a_i, b_j) = 1$. This means that even if two detections are identified with the same class, one can not affirm that they are the same object. This is due to the fact that there can be multiple different objects of the same class, e.g., in a real driving scenario many cars or pedestrians can appear. However, it is clear that if two detections have different classes it is more likely that they belong to different objects. Hence, we use the class information to provide evidence about the dissimilarity of detections and place it in $m^c_{a_i,b_j}$. The frame of discernment for the class evidence distribution is the set $\Omega = \{pedestrian, bike, car, truck\}$ of all possible classes of objects. The frame of discernment for detections' association is Ω_d and was described in Section 5.2.1. Hence, we propose to transfer the evidence from in Ω to Ω_d as follows:

$$\begin{aligned} m^c_{a_i,b_j}(\{1\}) =& 0, \\ m^c_{a_i,b_j}(\{0\}) =& \sum_{A \cap B = \emptyset} m^c_{a_i}(A) m^c_{b_j}(B), \\ & \forall A, B \subset \Omega, \\ m^c_{a_i,b_j}(\{1,0\}) =& 1 - m^c_{a_i,b_j}(\{0\}), \end{aligned} \tag{5.2.5}$$

which means that we fuse the mass evidences where no common class hypothesis is shared between detections in lists A and B. $m^c_{a_i}$ and $m^c_{b_j}$ represent the BBAs for the class hypotheses of detections in lists A and B. However, as we have no information about the possible relation of detections with the same class, we place the rest of the evidence in the ignorance hypothesis $\{1,0\}$.

5.2.2 Moving object tracking

Once we have performed moving object detection using lidar processing, the proposed approach obtains a preliminary description of the object position and object class encoded in m^c_{lidar}. Afterwards, using the regions of interest from lidar moving object detection and the camera based classifiers, a second evidence class distribution m^c_{image} is obtained. These two evidence distributions are combined using Equation 2.5.12 to form m^c_a.

Radar processing already provides a list of detections identified by their position and relative velocity. Following the method described in Section 5.1.2 we built the class distribution for radar detections m^c_b based on the relative speed of the radar targets. Both lists of detections' representations are processed in order to identify their associations and fuse their evidence distributions using Equations 5.2.2, 5.2.3 and 5.2.5.

Moving object tracking has to be performed in order to deliver the final output of a perception system. We follow the moving object tracking approach described in Section 4.1.1. Although we originally used this tracking approach for lidar data, we adapted this work to represent not only the lidar measurements but the fused representation obtained by our proposed fusion approach detailed in previous sections. Tracking interprets the fused representations' sequence by all the possible hypotheses of moving object trajectories over a sliding window of time. Generated object hypotheses are then put into a top-down process taking into account all the object dynamics models, sensor model, and visibility constraints. However, instead of searching in all the possible neighbour hypotheses, we use the class evidence distribution of each object detection to reduce the number of generated hypotheses by taking into account the class hypothesis with more mass evidence.

Hence, we include the evidence distribution for class hypotheses into the definition of edge relations in the graph $< V, E >$. Therefore, sibling edges need not only to share an spatial overlap, but to have a similar class. Two objects have similar classes if their classes belong to the same general set. Two sets of classes are defined as general: $vehicle = \{car, truck\}$ and $person = \{pedestrian, bike\}$. This new consideration is taken into account for parent-child siblings as well. Therefore, if an object has a high evidence mass in the hypothesis $\{car\}$, we only sample the possible hypotheses for *car* and *truck* objects. When the highest mass evidence is placed in a non-singleton hypothesis, such as *vehicle* the search space is expanded to include *car* and *truck* samples alike. The rest of the sampling and tracking process is kept unchanged.

Each time the tracking process is performed, the association between the object's

current state delivered by the object detection fusion approach is fused with the object description belonging to the corresponding track. This operation is done in the same way as in the dynamic fusion stage for the fusion approach at tracking level, described in Section 4.2.2. Doing this allows the complete solution to keep the object class information up-to-date each time new sensor data is available.

The tracking management process follows the same considerations presented in Section 4.1.1, where common operations between tracks are performed, i.e., track creation, track deletion, track splitting and track merging.

Our proposed tracking approach differentiates from the works like the one proposed by Vu (2009) in the core of the data association and moving object tracking. Whilst Vu (2009) uses a close to deterministic approach to perform the association and tracking, we use an evidential approach based on mass distributions to perform the data association. Our data association method is based not only in the kinetic nature of the object but in the appearance information obtained from the classification modules. This evidence distribution allows to reduce the search space to discover the object motion and shape models. Moreover, our approach keeps up-to-date the information of the object over time by performing an updating operation of the object class distribution when new sensor data is available. Hence, the final output of our DATMO solution is composed of a list of moving objects described by their kinetic information and by a set of all the possible class hypotheses represented by masses of evidence.

5.3 Experimental results

Using the sensor set-up described in Section 3.3, we gathered four datasets from real scenarios: two datasets from urban areas and two data sets from highways. Both data sets were manually tagged in order to provide a ground truth reference. Following the experimental scenario designed for our fusion approach at tracking level, we analysed the degree of improvement achieved by early inclusion of class information within the DATMO component of the perception problem. Moreover, we performed a comparison between the fusion approach at tracking level and the fusion approach at detection level using the same experimental scenarios.

In order to test our DATMO solution at detection level, we first performed SLAM for the lidar sensor measurements as described in Section 3.5.1 to detect the possible moving entities. Among the 2D position state for each object detection, we define the frame of discernment $\Omega = \{pedestrian, bike, car, truck\}$ for its evidence class distribution. Therefore, 2^{Ω} is the number of all the possible class hypotheses for each detection.

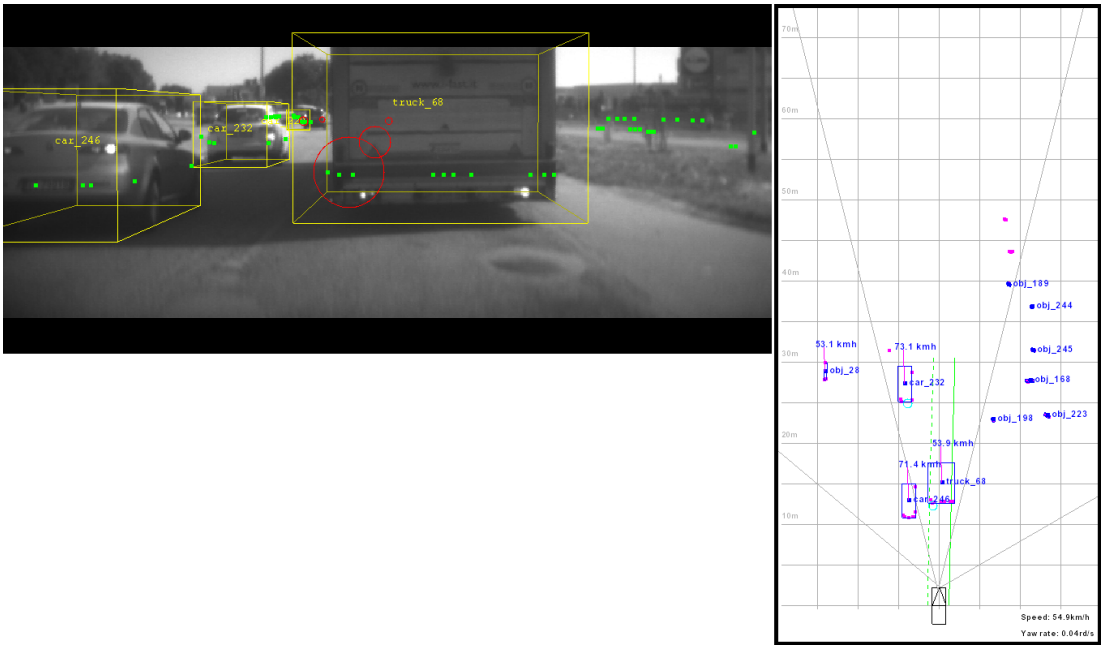


Figure 5.3: Results of the complete DATMO solution for urban areas. Several radar targets are discarded due to lack of support from the other sensor detections. The left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. The right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object.

Then, the object representations for lidar, radar and camera detections are extracted following the methodologies presented in Sections 5.1.1, 5.1.2 and 5.1.3, respectively. Once we obtained the object representations, we perform the fusion at detection level detailed in Section 5.2. Finally, the tracking of the fused object detections is performed following the approach presented in Section 5.2.2.

Figure 5.3 shows an example of noisy detections from radar sensor (red circles in the left image and cyan circles in the right image). The radar sensor detects several targets due to noisy measurements from moving obstacles like bushes, and from ghost objects. Even if these detections are kept after the data association, the confidence is considerable higher in the Ω hypothesis. Moreover, tracking process discards these detections as real because no further evidence is obtained and therefore are not considered as real moving objects of interest. In the same figure, the two cars in the left are not being detected by radar, but only by lidar. The preliminary class information from lidar detection and the classifier decisions from image classifiers allow the method to correctly identify the objects as cars. Moreover, the tracking process keeps the class information of these cars up-to-date when new sensor measurements corroborate the

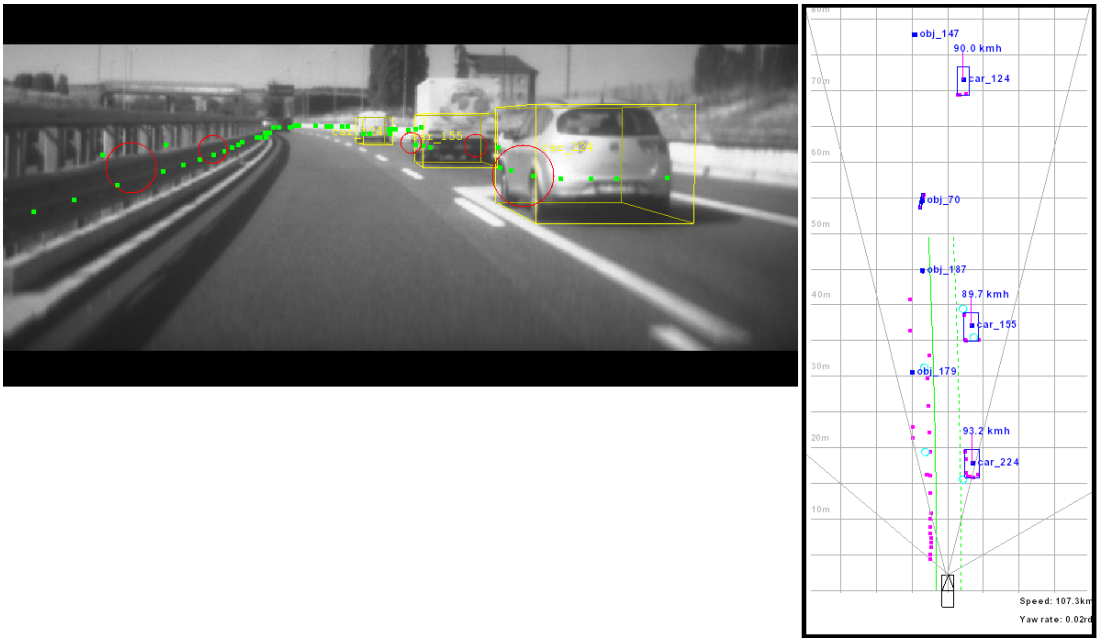


Figure 5.4: Results of the complete DATMO solution for a highway scenario. Vehicles at high speeds are detected. The left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. The right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object.

detected objects as moving entities of class *car*. The size of the bounding box is updated using the visible lidar measurements, the fixed-size class models and the lateral information from camera classification. The height of a bounding box is set according to the class of the detected object and to the result from camera classifiers.

Figure 5.4 shows three vehicles in front of the vehicle demonstrator. However, only two radar detections (from several spurious detections) are correct. Lidar and radar information correctly confirm the two closest vehicles, but only lidar processing perceives the farthest vehicle. In this situation, the lidar based detection and camera based classification evidence placed in m^c_{lidar} and m^c_{camera} correctly complement the information about the farthest vehicle. Moreover, the class of the moving objects is determined sooner than in the fusion approach at tracking level due to the early fused evidence about the class of objects. False moving object detections are not deleted at when fusion is performed but they are past to the tracking approach which will discard them after few mis-associations.

Figure 5.5 shows a cross road situation in an urban scenario. All the moving objects are detected but one car in the very front of the waiting line. Although this vehicle is

Figure 5.5: Results of the complete DATMO solution for urban areas. Several objects of interest are detected. Left side of the figure shows the camera image and the identified moving objects. Yellow boxes represent moving objects, green dots represent lidar hits and red circles represent radar detections. Right side of each figure shows the top view of the scene presented in the image. Objects classes are shown by tags close to each object.

sensed by radar, there is not enough evidence from lidar detection and camera based classification to verify its moving state. Moreover, this car is barely seen by lidar and few frames have passed to determine if it is moving or not. This car is considered a static unclassified object and appears in the top view. The car just behind this unrecognised car is as well considered static but it is identified and classified due to the previous detections that allow the determination of its moving nature.

Tables 5.1 and 5.2 show a comparison between the results obtained by the proposed fusion approach at detection level and our previous fusion approach at tracking level taking into account the erroneous classifications of moving objects. As in section 4.3 we use four datasets to conduct our experiments: 2 datasets from highways and 2 datasets from urban areas. We can see that the improvement of the fusion at detection level in highways with respect to the tracking level fusion is not considerable. However, in high-speed situations, the certainty about the moving vehicles is quite important. Hence, this small improvement is very useful for the final applications, such as continuous support systems. Urban areas represent a modern challenge for vehicle perception. The improvement of the fusion approach at detection level was considerable compared to our other fusion approach. Here, the richer representation of sensor

Table 5.1: Number of vehicle (car and truck) mis-classifications obtained by the fusion approaches.

Dataset	Moving objects	Number of vehicle mis-classifications	
		Tracking level	Detection level
Highway 1	110	6	4
		5.4%	3.6%
Highway 2	154	7	5
		4.5%	3.2%
Urban 1	195	20	10
		10.2%	5.1%
Urban 2	233	24	9
		10.3%	3.8%

Table 5.2: Number of pedestrian and bike mis-classifications obtained by the fusion approaches. Highway datasets do not contain any pedestrian or bike.

Dataset	Moving objects	Number of pedestrian mis-classifications	
		Tracking level	Detection level
Urban 1	52	11	6
		21.1%	11.53%
Urban 2	58	14	7
		24.13%	12%

Table 5.3: Number moving objects false detections obtained by the fusion approaches.

Dataset	Number of object false detections	
	Tracking level	Detection level
Highway 1	8	5
Highway 2	9	6
Urban 1	23	12
Urban 2	25	10

detections and the data association relations allowed the early detection of real moving vehicles.

Regarding the pedestrian classification results, we obtained similar improvements to those obtained for vehicle detections. The problem of small clusters detected by lidar as moving obstacles but without the certainty of being classified as pedestrians is mainly overcome by the early combination of class information from radar and camera-based classification. Moreover, the classification of moving objects (not only pedestrians) in our proposed approach takes on average less sensor scans than the compared fusion approach described in Chapter 4. This is due to the early integration of the knowledge about the class of detected objects placed in m_a^c and m_b^c, which is directly related to the reduced search space for the shape and motion model discovering process performed by the MCMC technique.

Table 5.3 shows the number of false detections obtained by the fusion at detection level and by the fusion approach at tracking level. In our experiments, a false detection occurs when a detection is identified as moving when it is not. These false detections occur due to noisy measurements and wrong object associations which are directly related with the lack of information in the detection, i.e. position, size, and class. The obtained results show that combining all the available information from sensor detections at detection level reduces the number of mis-detections and therefore provides a more accurate list of objects to the tracking process, which ultimately improves the final result of the whole perception solution.

5.4 Summary

In order to analyse how classification information can improve the early and final result of the DATMO component, in this chapter we have presented our multi-sensor fusion approach at detection level. First, we have described the object detection modules for lidar, radar, and camera. Afterwards, we have defined and built a composite object representation for each detected object. Classification information is extracted from the three sensors and is used to generate the appearance component of the object representation. Later on, we have described the general architecture of the fusion approach and detailed the data association component we use to find the detection associations before performing the evidential based fusion approach. Once we have obtained a fused object representation from the three different sensor detections, we used this representation to perform the tracking of moving objects. The tracking approach is based on the MCMC technique described in Section 4.1.1. However, we use the evidence distribu-

tion for class information to reduce the search space and reduce the number of frames needed to obtain a reliable shape and motion model for the moving objects. Finally, we have used the dynamic fusion approach described in Section 4.2.2 to update the object representation of the tracked objects every time new sensor data is available.

Several experiments were performed using three different sensors: lidar, radar, and camera. Following the scenario configuration used in the experimental section of Chapter 4, we used data sets from two real driving scenarios: highways and urban areas. The results showed considerable improvements compare to the multi-sensor fusion approach at tracking level. Improvements were achieved not only in the moving object classification but in the moving object detection. This means that when including classification information at early stages of the DATMO component, our proposed solution reduced the rate of mis-detections and mis-classifications; and therefore, the final result of the DATMO component is improved and enriched by the composite object representation. Although the obtained results were promising, it is interesting to test our contribution in a real-time application to observe how well it can be adapted to the constraints of a real intelligent vehicle perception system using on-line and off-line data. Therefore, these reasons motivated the selection of the fusion approach at detection level as the basis for the real perception application deployed inside the CRF demonstrator from the interactIVe project. The implementation of this perception application is detailed in the next chapter.

CHAPTER 6

Application: Accident Avoidance by Active Intervention for Intelligent Vehicles (interactIVe)

IN this chapter, we present an integration of our proposed object representation and multi-sensor fusion techniques as a part of a whole intelligent vehicle system solution. This solution is conducted in collaboration with the FIAT Research Center (CRF [1]) within the framework of the *interactIVe* European project [2]. The goals of this chapter are to present the implementation of our contributions inside a real vehicle application, and to show how these contributions can be adapted to fulfil real-time constraints. We focus on the perception system introduced in Chapter 3 where also was presented an overview of the *interactIVe* project. Inside the perception subsystem, we focus on the modules that represent where our contributions take place: the Frontal Object Perception (FOP) module and the Moving Object Classification (MOC) module. We build a composite object representation for each sensor detection from the early moving object detection stage. We use, as a basis, the fusion architecture at detection level presented in Chapter 5 to reduce the number of false detections. In order to obtain a real-time application and comply with the *interactIVe* requirements, we complement the basis fusion approach with the updating capabilities presented in Chapter 4. This adaptation provides a process that constantly updates the representations of the tracked objects while eliminating false tracks. An evaluation section is included to show the quantitative and qualitative performances of our perception solution inside the whole intelligent vehicle solution, implemented in the CRF vehicle demonstrator. We perform an on-line evaluation to cover the use cases presented in

[1] http://www.crf.it/en-US/Pages/default.aspx
[2] http://www.interactive-ip.eu

Table 3.1, and off-line evaluation to analyse the quantitative performance of our perception system. We finish the chapter with a summary of our participation inside the *interactIVe* project.

6.1 Fusion approach integration

In this section, we describe the integrations of our object representation and our multi-sensor fusion approaches inside two modules of the perception subsystem: Frontal Object Perception (FOP) and Moving Object Classification (MOC). Although schematically (from Figure 3.4) these two modules appear as separate entities, we follow the idea of performing SLAM and DATMO simultaneously. Therefore, the two modules are implemented as a unified module (FOP-MOC) where the detection, tracking, and classification of objects are performed simultaneously. The module takes input data from the three frontal sensors: camera, radar sensor, and lidar scanner as well as information about the dynamics of the ego-vehicle. The FOP-MOC module has been developed and evaluated in the CRF demonstrator vehicle, and thus, it makes use of the CRF front-looking sensor set described in Figure 3.7.

The goal of the FOP module is to deliver descriptions about relevant obstacles (e.g., location, speed) in the frontal area of the ego vehicle including stationary and moving objects which can be detected by the sensor platform. Also, the FOP module takes into account the far range objects which are more relevant to Continuous Support and Safe Cruise functions. MOC module aims to provide estimated information about class of moving objects detected by the FOP module. An object will be categorized into different classes: pedestrian (or group of pedestrians), bike, car, and truck. This information gives important values for the target applications when dealing with different classes of objects, for example: vulnerable road users like pedestrians and bikes.

In the following sections we will describe how we integrate some of the functions from our multi-sensor fusion contributions in Chapters 4 and 5 into the FOP-MOC module of the perception subsystem.

6.1.1 Modules architecture

According to the perception architecture, the FOP module takes sensor data inputs containing information about radar targets, camera images and raw lidar data. Additionally, dynamics of the ego-vehicle are also provided by another module named Vehicle State Filter (VSF). Simultaneously, the MOC module starts its processing after

the FOP module has finished and takes inputs from the FOP object list, lidar data and camera images. However, for practical reasons, we decide to implement MOC to run at the same time with FOP. In this way, our common module performs simultaneously object detection, tracking (FOP), and classification (MOC) of moving objects.

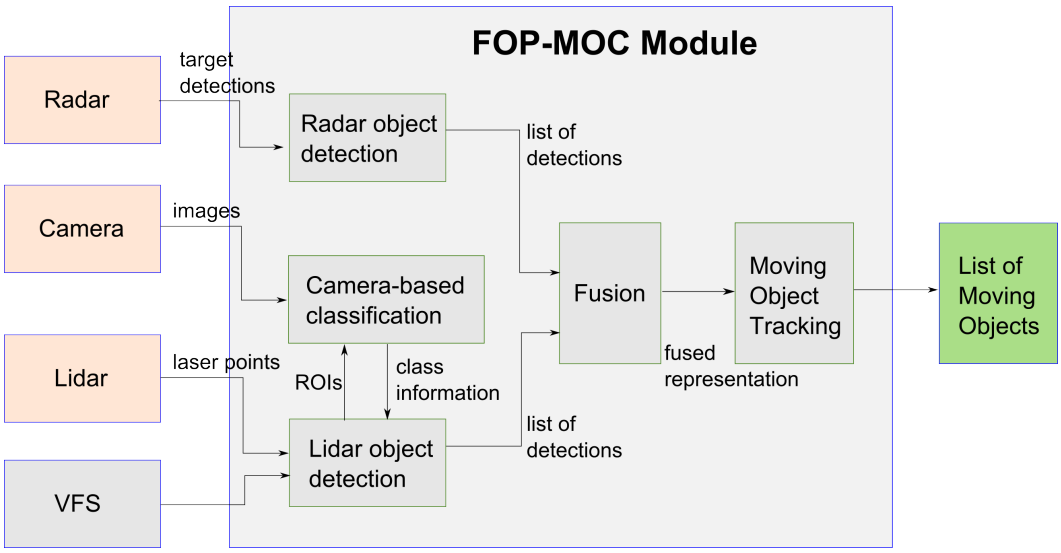


Figure 6.1: Integrated FOP-MOC module architecture.

The detailed data processing flow of the integrated FOP-MOC module is shown in Figure 6.1, which is comprised of five main components:

- Lidar object detection. In this stage, a grid-based fusion approach is used to incrementally build a local map surrounding the ego-vehicle. Based on the constructed grid map, static objects and moving objects are detected when new measurements are received.

- Camera-based classification. Due to the goals of the target applications, our module has to focus on several classes of objects on the road, for example: other vehicles (cars and trucks) and vulnerable users (pedestrians and bikes). In this stage, dedicated object classifiers (also known as detectors) are designed to identify a particular object class from camera images. An attention-focused fusion method is employed using the targets detected by radar and lidar sensors to locate regions of interests (ROIs) in the image and hence significantly reduce the search space. Output of this stage is a list of image objects together with their visual class likelihoods.

- Radar object detection. In this module radar targets are retrieve from the sensor output to built a radar-based list of moving object detections. These detections are used as a confirmation factor to identify moving objects detected by lidar.

- Fusion. In this stage, a unified fusion process takes place after the single-sensor object detection processes to confirm the objects detected by different sensor modules (i.e., radar, lidar and camera). Object class (MOC information) is also decided based on the fusion of their preliminary class likelihoods estimated individually by lidar, radar, and image processing components. The architecture shown in Figure 6.1 is a specialized version of the fusion architecture at detection level shown in Figure 5.2, here the detection modules and classification modules are driven by the perception systems constraints, specially the real-data requirement.

- Moving Object Tracking. Moving objects are tracked using a model-based approach where object velocity, geometry, and a preliminary classification are estimated at the same time. Output of this stage is a list of objects represented with their location, geometry, class, and dynamics. The preliminary class likelihood value for each moving object is estimated based on its size and its moving speed.

According to the Perception Platform architecture shown in Figure 3.4 the FOP and MOC modules are required to gather, process, and deliver the perception output in a time window of less than 75ms. In the following subsections we describe in detail how we implement the contributions presented in Chapters 4 and 5 inside the FOP-MOC module.

Lidar-based object detection and moving object tracking

We have to be aware of the possibility that the objects appearing in front of the ego-vehicle can be either static or moving objects. In order to correctly map the environment, reduce the tracking complexity, and focus on the moving objects of interest, we would like to distinguish static objects from moving ones and treat them separately according not only to their dynamic nature but also to their class.

In order to detect static objects from the raw lidar data which is also known as the mapping problem, we employ our proposed grid-based fusion approach described in Section 3.5.1. Using the ego-vehicle dynamics (provided by VSF module), discrete lidar scans are incrementally integrated into a local occupancy grid map M representing the environment surrounding the ego-vehicle. A high value of occupancy grid indicates that the cell is occupied and a low value means the cell is free. As a remark from our previous work, one advantage of using grid-map representation compared to other approaches using feature-based or direct representation is that the noise and sparseness of raw lidar data can be inherently handled and at this low-level fusion.

Based on the constructed grid M, when a new lidar data measurement is received, a static object can be detected if it appears at occupied regions. Since a static object can be of various size and shape, it is then extracted in the form of contour points or bounding box of measurements depending on the target application. After the moving parts of the map M are identified, a new grid D is built comprising only of dynamic parts.

For moving objects, we can follow a traditional detection-before-tracking approach using the local grid map to detect objects when they enter the object-free region. Then, the list of detected objects is passed to a tracking module where two main tasks are addressed: data association and track management.

Within the European Commission initiative to build safety automotive solutions, previous projects such as the PREVENT/ProFusion project, implemented a tracking approach based on the idea of using a Multiple Hypothesis Tracker (MHT) for the data association and an adaptive Interactive Multiple Models (IMM) algorithm for the track filtering. However, this approach has two known problems as described in (Vu and Aycard, 2009). First, due to the inherent discreteness of the grid and threshold functions, moving object detection at one time instant usually results in ambiguities with missed/false detections and objects can be split into several segments that make data association for tracking sometimes very difficult. Second, due to the fact that the lidar sensor only sees part of an object, object extraction in this way does not always reflect the true geometry of the object which severely affects the accuracy of the tracking result.

In order to overcome the previously mentioned drawbacks, we highlight the fact that the number classes of moving objects of interest is quite limited and fortunately they can be represented by simple geometric models, for example: rectangles for vehicles and bicycles and small circles for pedestrians. Therefore, we propose to use a model-based approach which formulates the detection and tracking of moving objects as a batch optimization problem using a temporal sliding window over a fixed number of data frames as was described in Section 4.1.1. The detection process that is performed at a single sensor frame (based on the occupancy grid map mentioned above) is used as a coarse detection that provides bottom-up evidence about potential moving objects present in the D grid. Since these evidences are actually visible parts of the objects, they are used to generate hypotheses of actual objects using all possible object models. Object hypotheses generated from all frames are then put into a top-down process taking into account all the object dynamics models, sensor models, and sensor visibility constraints. This leads to an optimization problem that searches for

a set of trajectories of moving objects best explaining the measured data. In order to reduce the search space, our method only focuses on the neighbour objects of similar class, as was described in Section 5.1.1. The optimal solution is found by a very efficient MCMC-based sampling technique that can meet the real-time requirement, it can find the solution in several milliseconds. This new approach has two advantages. First, the detection and tracking of objects over a number of data frames are transparently solved simultaneously which significantly reduces the ambiguities that might be caused by detection at a single frame. Second, using a model-based approach, object's geometry is estimated and updated, moreover the class of the object is also estimated simultaneously based on the discovered geometry. These two advantages help to improve the overall tracking results as will be shown in the experimental section.

The output of the described simultaneous detection and tracking of moving objects includes a list of objects with their locations, dynamics, and estimated geometry. At the same time, and for the purpose of the later computation of MOC information, for each object class of interest, we estimate a likelihood value of a returned object belonging to that class based on its geometry and moving speed. For example, an object with a large size and high speed will have less chance to be a pedestrian and more chance to be a vehicle. A rectangular object with a longer length is more likely to be a truck than a car. In order to do this preliminary classification we use the BBAs described in Section 5.1.1 and Section 5.1.2. In order to have a single BBA representing the evidence of the class of the object we compute Yager's rule of combination to calculate the lidar mass distribution m_l over the class frame of discernment Ω_c, as was described in Section 5.1.1. Finally, the result provided by lidar processing can be seen as a list of moving objects represented by their kinematic and appearance information, i.e., position, geometry, and class evidence. Moreover, the position and geometry can be seen as a regions of interest which are used by the camera-based classification process to focus on specific areas of the camera image.

Radar targets

Experimentally, we have seen that the targets provided by radar sensor are mostly ghosts objects or noise from non-interesting moving objects, such as bushes. In order to speed-up the sensor processing we do not perform target tracking, instead we considered the targets as sensor detections. These sensor detections are represented using the composite object representation described in the previous chapter. Therefore, we have the kinematic and appearance information for each radar target. This list of targets is fused with the lidar-based list of moving objects using the fusion at detection level approach described in Section 5.2. The idea in doing this is to use the lidar in-

formation as a reference and the radar information as complementary data to confirm lidar detections. However, as was described in Section 5.2, we do not eliminate the target detections that are not associated to any lidar detection, instead we let the tracking process decide if these detections belong to be real moving objects or not.

Camera-based Object Classification

In order to detect objects from images, we follow the most popular approach: a sliding-window technique where the detection window is tried at different positions and scales. For each window, visual features are extracted and a classifier (trained off-line) is applied to decide if an object of interest is contained inside the window. In general, the choice of image representation and classification method decides the performance of the whole system.

We based our approach on the visual object representation mentioned in Section 3.5.3, also known as histograms of oriented gradients (HOG) which has recently become a state-of-the-art feature in computer vision domain for object detection tasks. We used the Sparse-HOG visual descriptor presented in Section 3.5.3 to represent the possible sections of the image containing objects of interest. We used 6 histogram bins for all object classes, 9 blocks for pedestrian, 7 blocks for bikes, 7 blocks for car, and 7 blocks for truck.

Generating a S-HOG descriptor per image patch per size and scale variation involves considerable computational time. In order to meet the real-time requirement we need to speed-up this process. In order to accelerate the S-HOG descriptor computation, we employed the technique introduced by Viola and Jones (2001b), which proposed the idea that rectangle-shaped features can be computed very rapidly using an intermediate representation of the whole image, also known as *integral image*. We compute and store an integral image for each bin of the S-HOG (resulting in 6 images in our case) and use them to efficiently compute the S-HOG for any rectangular image region, which requires only 4 x 6 access operations to the image.

Given computed features, the choice of classifiers has a substantial impact on the resulting speed and quality. To achieve a suitable trade-off, we chose the discrete Adaboost method, a boosting-based learning algorithm (described in Section 10). The idea of a boosting-based classifier is to combine many weak classifiers to form a powerful one where weak classifiers are only required to perform better than chance hence they can be very simple and fast to compute.

For each object class of interest, a binary classifier is pre-trained to identify object

(positive) and non-object (negative) images. For the off-line training stage, positive images are collected from public datasets (such as the DAIMLER dataset) or manually labelled datasets containing objects of different viewpoints, for example: pedestrian (frontal, profile), car (frontal, rear side), truck (frontal, rear side). They are all scaled to have sampling images of the same size for each object class: pedestrian: 32x80 pixels, bike: 50x50, car: 60x48 pixels, and truck: 60x60 pixels. Negative samples are generated randomly from images which do not contain objects of interest. S-HOG features are computed for all samples which are then used for training classifiers.

Several object classifiers were trained off-line to recognized the objects of interest: pedestrian, bike, car, and truck. Following the classification process described in Section 10, the training process starts where each training sample is initially assigned the same weight and iterates for a fixed number of times. On each round, a weak classifier is trained on the weighted training data and its weighted training is recomputed. The weights are increased for training samples being misclassified so that the weak classifier is forced to focus on these hard samples in the next step. The classification is based on the sign of the weighted sum over individual learned weak classifiers. In our implementation, decision trees are used as weak classifiers in this boosting scheme.

Final classifiers for each object class obtained after off-line training are used for the on-line object detection stage in a sliding-window scheme. Detection time is affected by both phases of feature extraction and classification. Thanks to the use of the integral image, the feature extraction step is very fast only taking about 10ms or less. Likewise, the classification time is fast as well, taking only about 2ms per 100 samples. For an input image of size of 752x250 pixels, there are several thousand windows to check and the whole detection time is about 70ms for each object class.

Although the image object detection process is quite fast, we still can accelerate this process to meet the time requirement of the FOP-MOC module (less than 75ms). Instead of searching for the whole input image, we make use of information about targets detected by radar sensor and lidar processing module described above to focus on specific regions of interest (ROIs) over the camera image. Besides, thanks to the lane information provided by the camera, we can compute the homograph to transform coordinates of radar and lidar targets onto the image to calculate ROIs. In this way, the number of sliding windows can be then reduced to several hundred to make the computational time of the whole detection process between 20 and 30ms.

Our image-based object detection approach can directly obtain a class likelihood for each correct object detection. In order to do so, it estimates the class likelihood based on the number of detections around the object location. Generally, the greater the number

of positive windows (containing an object of interest), the greater the likelihood is that the object belongs to that class. We follow the camera based classification described in Section 4.1.3 to built the evidence distribution for the camera-based classification m_c over the frame of discernment $\Omega = \{pedestrian, bike, car, truck\}$.

Fusion implementation

At this stage, a unified fusion process takes place to verify the list of objects detected by different sensors (e.g., radar, camera, lidar) in order to decide the final FOP-MOC output. Since sensors have different fields of view, the fusion is performed only in the overlapping region of the common coordinate system. However, non fused detections are passed to the tracking process to deal with them in the next frame, as was proposed in Section 5.2.2. Moreover, as different sensors have different characteristics, the fusion aims to make use of the complementary information of these sensors to improve the overall object detection and classification provided by individual sensors. Additionally, conflict evidences can be used to reduce the number of false positives, mis-classifications, and spurious tracks.

Our fusion approach is based on the DS theory and is powered by the early fusion capabilities of our fusion approach at detection level presented in Chapter 5 and the two-stage updating feature of the fusion approach at tracking level presented in Chapter 4. It takes, as sources of evidence, individual object detection lists provided by all the sensors: lidar, radar, and camera. For each object, its state includes information about its location, shape, size, and velocity together with preliminary object classification provided by the lidar & radar and the camera sensor. Using the DS theory we are able to represent evidence about these object features coming from different sensor detectors, and their classification likelihood into a common representation: a mass evidence distribution. The proposed fusion process relies on three main parts: the target fusion from the lidar and radar; the instantaneous fusion, obtained from the combination of evidence provided by individual sensors at current time; and the dynamic fusion, which combines evidence from previous times with the instantaneous fusion result. The fusion mechanism used to combine the sources of evidence was extracted from our fusion contributions presented in Sections 4.2 and 5.2. This mechanism allows us to give more support to common hypotheses and use complementary evidence by managing situations with different levels of conflict without getting counter-intuitive results. These situations usually appear when sensors with low reliability are used, their evidence is noisy or contradictory, and when the demonstrator is placed in cluttered scenarios. The fusion approach inside the FOP-MOC module is

described in Chapter 4 and Chapter 5 which represent our main contributions to the perception subsystem.

Given that the performance of the individual object detectors varies according to the type of sensor and their specifications, we included two uncertainty factors into the rule of combination: sensor reliability and sensor precision to classify certain types of objects. The final state (location, shape, size, velocity and classification information) for each object is selected as the hypothesis with the highest evidence value after the dynamic fusion is performed. By doing this, the final result comprises the most sensor capabilities to detect specific features of the object. For example, a camera sensor provides a better approximation of an object's width, a radar sensor can give a direct measurement of relative speed, and a lidar sensor can give the distance and direction to a moving object providing also more accurate measures of object's shape and size when it is available. Another example occurs when in cluttered urban areas, which are a common scenario where image-based classifiers capabilities help to classify a group of pedestrians correctly where usually lidar processing is not able to. The output of this stage is a list of objects, in the frontal area of the vehicle, described by a composite representation. This means, each moving object representation contains all information about the object's kinematic (location and dynamics) plus the object's and appearance (geometry and classification information from the fusion process). Detailed descriptions of our fusion methods applied to the CRF vehicle demonstrator are presented in (Chavez-Garcia *et al.*, 2012), (Chavez-Garcia *et al.*, 2013) and (Chavez-Garcia *et al.*, 2014). We implemented the fusion approach based on the fusion architecture at detection level. This implementation includes an early detection fusion stage from lidar, radar and camera which a continuous updating process carried out by the two main modules of the fusion at tracking level: instantaneous fusion and dynamic fusion.

This version of the fusion approach allows us to be able to deal with noisy detections at an early stage without discarding them until new evidence arrives to verify their status. Classification information plays an important role in this fusion approach due not only to its non deterministic nature, but to its ability to improve the data association process at detection level. Moreover, as was detailed in Section5.2.2, classification information can improve the sampling technique for model-based tracking, and enrich the object representation from the beginning of DATMO component to the final result of the perception task. In the experimental section we will show how the detection and classification of objects of interest is overcome by using our proposed DATMO solution: sensor data processing, moving object detection, moving object tracking, object classification, and a fusion implementation of the fusion architecture at detection level

adapted to meet the requirements of the project application.

6.2 Experimental evaluation

According to the *interactIVe* project goals, the testing & evaluation process for the FOP-MOC module should address the following points:

- Computational time. The module's computing time should be checked against the constraint required by the reference platform.
- Qualitative performance evaluation. General functionalities of the whole module (i.e., object detection, tracking and classification) should be checked to see if they are working as expected.
  - Improvement of the fusion process. We are interested in seeing how fusion processes can help improve the performance quality.
- Quantitative performance evaluation. This procedure should be based on evaluation metrics to assess the accuracy of each function:
  - Detection. The accuracy is measured in terms of correct detection and false detection.
  - Classification. the accuracy is measured in terms of correct classification, missed/false classification.
  - Tracking. The accuracy is measured in terms of object location, velocity, and geometry compared with the real values. This requires the ground-truth data for assessment.

Since specific road environment characteristics can affect the module's performance, apart from test track scenarios designed to measure the module's performance in controlled situations, the FOP-MOC module needs to be tested in different real environments: highway, urban areas and rural roads. The speed of the ego-vehicle demonstrator varies according to the test scenario, e.g., low speeds for urban areas and high speeds for highways ($> 60km/h$). Also, the module should provide accurate and on time detection of critical and non-critical objects. A qualitative evaluation is performed on-line and off-line. The quantitative evaluation is performed off-line using several datasets gathered from the CRF vehicle demonstrator.

6.2.1 Real time assessment

Instead of being processed at different levels as originally designed (see Figure 3.4), the integrated FOP-MOC module is triggered to run as a unified module from the beginning of the FOP module to the end of the MOC module inside the Reference Perception Platform (RPP). This concatenation of modules allows us to have a gain of total time up to 75ms (per 100ms of one RPP cycle) which nevertheless is still a challenge for the whole sensor data processing.

Our decision of building the FOP-MOC as a unified module allows us to have a computational window of 75ms to perform our whole DATMO solution. Based on the running time statistics of our module, its average computing time is about 40ms which fulfils the requirement of the designed platform. However, in some cases, due to issues with the synchronization process, the FOP-MOC module does not provide a complete output within few cycles to the RPP. In these cases, the processing time scales up to 60ms which is still below the time limit.

6.2.2 Qualitative evaluation

The purpose of this assessment is to verify, from the final user point of view, the general functionalities of the FOP-MOC module in terms of object detection, tracking and classification. In the following, we will show some results obtained from different test scenarios. Output provided from the FOP-MOC module is checked with the camera video to see if all the functions are working as expected.

Figure 6.2 shows two scenarios for car detections on test track scenarios. In the first situation, the vehicle demonstrator is approaching a stationary vehicle. In the camera view, we can see that the target car is detected and well classified. Although it is seen by all sensors: radar (red circle), lidar (green dots), and camera (yellow box), the lidar only sees the rear part of the car. However, the preliminary classification provided by the lidar processing along with the camera-based classification allow the correct detection of the object's class. In the second situation, the ego-vehicle is following a moving car, which is seen by all the sensors and is correctly classified as a car. In this situation, when the target moves, the lidar is able to estimate the target size which also helps the correct classification. The accuracy of the lidar tracking algorithm can be verified by comparing the lidar-estimated car speed with the speed provided by the radar sensor and the speed of the vehicle demonstrator. Radar sensor only provides Doppler velocity of the target and no information about the target moving direction. However, thanks to the lidar tracking module, the car moving direction and its geometry are well

estimated which is very important information for the danger situation assessment in the final application.

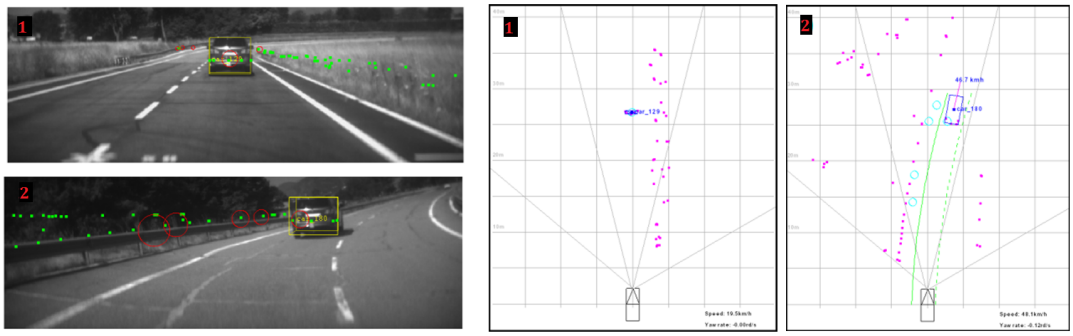


Figure 6.2: FOP-MOC module output: detection, classification and tracking of cars on test track scenario. The test tracks scenario is part of the CRF facilities.

Regarding pedestrian detection, Figure 6.3 shows two examples of pedestrian detections on the test track. In these examples, the vehicle demonstrator is not moving. In the first situation, two pedestrians are crossing each other in the frontal area and in the second situation, two pedestrians are moving closely towards the ego-vehicle. In both cases, we observe that radar detection of the pedestrians is not fully reliable in particular for distances above 30m. On the other hand they are well detected and tracked by the lidar. However, only the camera is able to provide good class information of objects. Two pedestrians in the first test are well recognized and the final target in the second test is correctly classified as a group of pedestrians thanks to the image classification module.

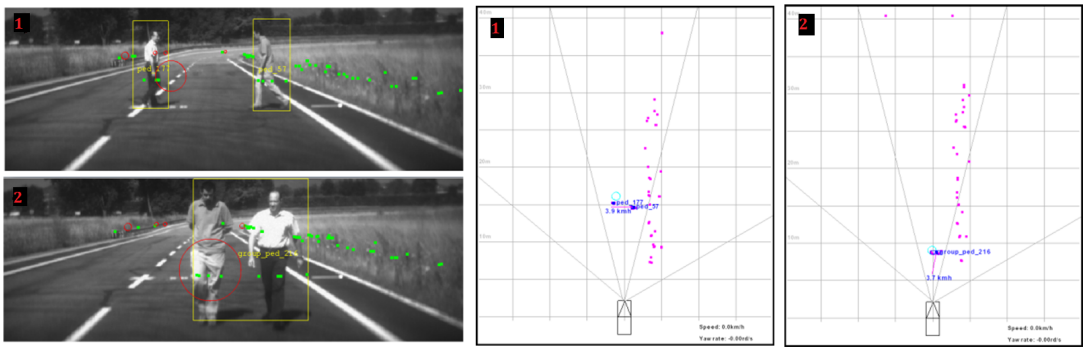


Figure 6.3: FOP-MOC module output: detection, classification and tracking of pedestrians on test track scenario. The test tracks scenario is part of the CRF facilities.

Figure 6.4 shows two output examples of the FOP-MOC module in real driving scenarios: highway and urban areas. Both scenarios are considered as high traffic scenarios due to the large number of moving objects around the vehicle demonstrator. In both scenarios, all vehicles, including oncoming ones, are well detected, tracked and

correctly classified: several cars and two trucks in the highway; and several cars, one truck and one pedestrian in the urban area. Additionally, static objects (such as barriers) are also reported and correctly identified as static obstacles using the static map built in the early stages of our proposed solution. In the top view of figures examples, moving objects are distinguished by their speeds. In addition, their moving directions are well estimated thanks to the model-based tracking module which uses the composite object representation to calculate the moving object estimates. Note that in the early fusion stage, the radar Doppler velocity information helps to improve the target speed estimated by the lidar after its moving direction is known. Once more, the class of the object is improved by the fused information from the three different sensors providing a more reliable class information in the form of a class distribution. The target applications use this class distribution to decide the final class of the object.

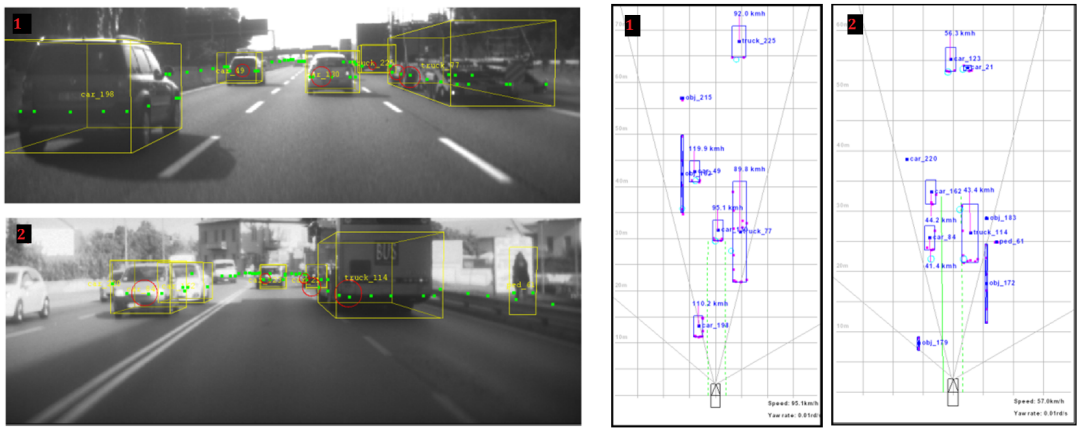


Figure 6.4: FOP-MOC module output: detection, classification and tracking of objects of interest in real scenarios. First case: highway. Second case: urban areas.

Through several off-line and on-line tests on test tracks and in real driving scenarios, it is qualitatively shown that the FOP-MOC module, comprised of detection, tracking and classification, performs well. Moreover, it is interesting to observe how our proposed fusion process helps to make use of the best characteristics from different sensors into the final perception output. A common moving object is represented by several pieces of information: location, geometry, object class, speed, and moving direction that cannot be provided by only one individual sensor.

6.2.3 Quantitative evaluation

In order to evaluate the performance of our FOP-MOC module, we have manually created a ground truth data using four different scenarios: test track, highway, rural road, and urban area. The quantitative evaluations focus on the detection and classification

functions since they are more critical to the target application developed for the CRF vehicle demonstrator.

We choose four typical driving scenarios from the available dataset and performed a frame-by-frame evaluation. For each data frame, we manually labelled the moving objects of interest (e.g., pedestrian, bike, car and truck) identifiable by human eye from the camera video. Whilst running the FOP-MOC module, for each object, we will count in how many frames it is correctly detected and classified. The number of wrong-detections and wrong-classifications (false positives) were also counted.

Table 6.1 summarizes the results collected after testing the FOP-MOC module with the data from the four different scenarios: highway, urban area, rural road and and test track. For clarity sake, the number of correct and false detections and classifications are also represented by percentages. Four objects of interest were taken into account: pedestrians, bikes, cars and trucks. However, not all of the objects of interest appear in all the test scenarios, e.g., pedestrians and bikes are not present in highways. Correct detections are well identified moving object detections, which are counted according to type of object of interest. False detections are considered as moving object detections that do not represent a real moving object. Correct classifications are well classified objects from the correctly detected moving objects. False classifications are self-explanatory.

From Table 6.1, we can see that in all the tests performed, for all considered objects of interest, high detection and classification rates were achieved with relatively low false positives.

In the test track scenario, where only one car or a few pedestrians are present, the detection and classification rate of pedestrians and cars are nearly perfect (96-100%). This scenario does not contain many of the common driving situations, such as several moving vehicles, pedestrians, crossing lines, and high traffic dynamics. However, this controlled scenario allows us to test specific components of the FOP-MOC modules, e.g., pedestrian and vehicle classifier, moving vehicle tracking.

Bikes rarely appear in any of the available test data, that is the reason why their count number is so low even in urban areas. However, when these objects appear, the FOP-MOC module identifies almost all of them.

In highway scenarios, the detection rate of vehicles is also very good: car (97.8%), truck (96.4%) where the missed detections are due mainly to inherently noisy and cluttered data (e.g., lidar impacts on the ground). The large size of the truck makes the truck detection not as accurate as car detection since it is sometimes confused with the

barrier. The false detection rate (2.2%) is due mainly to the reflection in raw lidar data which creates ghost objects and the noisy radar target detection. However, the fusion approach allows the ability to obtain a highly correct classification rate for both cars or trucks whilst keeping a very low false classification rate.

In urban scenarios, vehicle detection and classification is still high, considering the increased number of moving obstacles and the cluttered environment. However, the false detection rate is higher than the one obtained in the highway scenario. This increase is due to the highly dynamic environment and to the reduced field of view in high traffic situations. Moreover, the pedestrian false classifications commonly appears when the classifiers mis-classify traffic posts as pedestrians. These mis-classification situations suggest the construction of more robust classifiers or the implementations of a more discriminating visual descriptor.

Rural roads are a less cluttered driving scenario. Although several moving objects of interest may appear, high traffic dynamics are not present. Besides, road barriers are not a common landmark in this scenario, which provides an interesting test scenario to evaluate the entire perception platform. The false classification rates obtained in this scenario are higher than the ones obtained from the other three scenarios. This is due to the increasing number of natural obstacles, such as bushes and trees. The common object false classifications are due to false moving objects (mainly bushes) preliminary classified as trucks or pedestrians. One solution could be to implement a dedicated classifier to discard this type of obstacles.

6.3 Summary

In this chapter, we detailed the application of our contributions inside the *interactIVe* project. We described how our composite object representation as well as our fusion approaches, at detection and tracking level, were implemented as the core of the Frontal Object Perception - Moving Object Classification (FOP-MOC) module. This module is placed inside the perception subsystem of a whole intelligent vehicle solution. We can see this perception subsystem as a complete SLAM+DATMO component. We employed the sensor set-up inside the CRF vehicle demonstrator to gather datasets and to test the final Continuous Support and Safe Cruise applications both off-line and on-line. The final evaluation and the final on-line demonstrations empirically demonstrate the great impact of considering classification information not only as an aggregate but as a key element of the object representation. Also, experiments have shown that this key element needs to be constantly updated to represent the evolving evidence of the

Table 6.1: Quantitative results of the FOP-MOC module for four different scenarios: highway, urban area, rural road and test track. Four objects of interest are taken into account: (P) pedestrian, (B) bike, (C) car, and (T) truck.

Scenario	Total objects				Detections					Classifications							
					Correct				False	Correct				False			
	P	B	C	T	P	B	C	T	all	P	B	C	T	P	B	C	T
Highway	0	0	702	281	n/a	n/a	687	271	22	n/a	n/a	669	251	0	0	4	0
					n/a	n/a	97.8%	96.4%	2.2%	n/a	n/a	95.2%	89.3%	0%	0%	0.5%	0%
Urban	65	7	619	97	57	6	580	88	17	57	6	570	78	9	1	6	5
					87.6%	85.7%	93.6%	90.7%	2.1%	87.6%	85.7%	92.0%	80.4%	13.8%	14.2%	0.9%	5.1%
Rural	9	0	68	6	9	n/a	62	5	9	9	n/a	60	5	3	0	5	2
					100%	n/a	91.1%	83.3%	10.8%	100%	n/a	88.2%	100%	33.3%	0%	7.3%	33.3%
Test track	248	0	301	0	247	n/a	300	n/a	1	240	n/a	300	n/a	0	0	0	0
					99.6%	n/a	100%	n/a	0.1%	96.7%	n/a	100%	n/a	0%	0%	0%	0%

object class. Moreover, regarding the fusion approach at detection level, this evidence of the class of an object can be used to accelerate and improve the model-based tracking process.

From the initial tests and evaluation process, the FOP-MOC module has shown to perform well providing reliable outputs in term of detection and classification while maintaining the tight computational time requirement of the whole Perception Platform. These good results are possible due to the introduction of new sensor data processing algorithms as well as the sensor fusion approaches at different levels of the DATMO component. We focus the evaluation of the FOP-MOC module on the detection and classification of moving objects of interest. However, in order to test the tracking results with more accurate precision, ground-truth data for the testing scenarios are required. This data-set is not yet available for our datasets due to the technical complexity. Nevertheless, the construction of this dataset is considered an ongoing process.

CHAPTER 7

Conclusion and Perspectives

IN this dissertation, we have reviewed the problem of intelligent vehicle perception. We have analysed this problem taking into account its two main components: simultaneous localization and mapping (SLAM) and the detection and tracking of moving objects (DATMO). We have studied the traditional solutions for these tasks and have focused on the DATMO component where our main contributions take place. After reviewing the state-of-the-art approaches, we have decided to work with several sensors as a first approach to improve the perception output. The three main sensors that were used to define, develop, test and evaluate our contributions were: lidar, radar, and camera. We have proposed the use of classification information as a key element of a composite object representation, where not only kinetic information but appearance information plays an important role in the detection, classification and tracking of moving objects of interest. We have analysed the impact of our composite object description by performing multi-sensor fusion at two levels inside the DATMO component. At the tracking level, we improve the common late fusion approach by integrating classification information to constantly update the evidence distribution of the class of moving objects. At the detection level, we have improved the detection by considering an evidence distribution over the different class hypotheses of the detected objects. This improvement directly reduces the number of false detections and false classifications at early levels of the DATMO component. Moreover, the tracking stage benefits from the reduction of mis-detections and from the more accurate classification information to accelerate the tracking process by limiting the search space of our tracking approach.

7.1 Conclusions

In Chapter 2, we have formally presented the perception problem for intelligent vehicle systems. Here, we have focused on the wide number of tasks inside the DATMO component. We have reviewed the state-of-the-art approaches for the moving object detection, classification and tracking of moving objects. Moreover, we have defined the problem of information fusion for multi-sensor systems; and we have reviewed the most common techniques to perform fusion in the vehicle perception field. This review motivated us to choose the Transferable Belief Model (TBM) to represent and include the classification information inside our object representation. The TBM plays an important role inside our fusion approaches due to its built-in mechanisms to fuse information. Moreover, we have presented the sensor processing modules for the three different sensors we use. These modules are the initial processes inside our two proposed multi-sensor fusion approaches.

In Chapter 4, we described our fusion approach at tracking level. Here, we use a late fusion approach architecture to combine information from different sensor trackers and classifiers. However, we put our evidential class representation as an extension of the late fusion approach to improve the final moving object classification. This fusion approach takes into account the uncertainty from the object classifiers and from the sensor processing modules to provide a more accurate hypotheses distribution over the class of the detected objects. Our approach performed fusion using two stages: the instantaneous fusion combines information from the current sensor measurements; and the dynamic fusion combines information between the instantaneous fusion result and the previous combinations. We integrated our fusion approach at tracking level inside a whole perception system to test it and evaluate it using real data from a vehicle demonstrator. The results showed an improvement in the moving object classification with respect to the single sensor trackers and classifiers. Our fusion approach at tracking level took into account the limitations of the related works by: including classification information not only as an aggregate; proposing different mechanisms to obtain the class of the objects using the available sensor information; updating the moving objects state using not only the information at current time, but from previous sensor measurements; and keeping an evidence distribution of the possible object class hypotheses until a decision module needed a final object classification decision.

The results obtained by our fusion approach at tracking level motivated the proposal of a fusion approach at detection level. Classification information was a very useful asset for the fusion at tracking level. Therefore, we wanted to analyse the improvement of the final perception result by introducing the composite object repre-

sentation in the early levels of the DATMO component: the moving object detection task. In Chapter 5, we described our fusion approach at detection level taking into account the three different sensors we mentioned above. More problems were found at this level. We proposed several classification modules to obtain a preliminary evidence class distribution from the three different sensors. A data association approach was proposed to find the object detection relations before the fusion was performed. Several object similarity functions were proposed to represent the potential relations between the detected objects from different sensors. An extension of the model-based tracking method proposed in Chapter 4 was introduced to take into account the fused object representation obtained from our fusion approach at detection level. We took the two-stage process from our fusion approach at tracking level to keep the objects state up-to-date. Several experiments were performed in order to analyse the improvement of our fusion approach at detection level. We performed a comparison between our two fusion approaches to review the improvement from introducing the classification information at early levels of the DATMO component. Experimental results showed promising evidence that by including our object representation, data association algorithm and fusion techniques at detection level, we can obtain better results. These results improve the perception outcome not only in terms of moving object classification, but in terms of reducing the number of false moving object detections, which directly benefits the final result of the perception task.

Finally, in Chapter 6 we present the implementation of our contributions as a part of a real intelligent vehicle application. Taking as the basis the fusion approach at detection level, we integrate the modules presented in Chapter 4 and Chapter 5 inside the perception subsystem of the interactIVe European project. Specifically, we were in charge of the Frontal Object Perception (FOP) and Moving Object Classification (MOC) modules. In this chapter, we described the general architecture of the interacIVe project and the specific components of our FOP and MOC modules. These modules can be considered as the core of the perception sub-system of a real Lancia Delta vehicle demonstrator, described in Figure 3.6, where three main frontal sensors are used to sense the environment: lidar, radar, and camera. Extensive experiments on controlled and real driving scenarios empirically demonstrated the positive performance of our modules for moving object detection and moving object classification. The reliability of our module results were quantitatively and qualitatively tested in on-line and off-line demonstrations. Whilst the final application obtained interesting results on real time tests, it opened new challenges for more accurate and improved safety applications.

In summary, we have addressed the challenge of detection, tracking, and classifica-

tion of moving objects inside the perception problem. We proposed a composite object representation and two different multi-sensor fusion approaches to integrate classification information inside the DATMO component. We have empirically shown how the inclusion of our different approaches improved the final result of a real perception system.

7.2 Perspectives

Although the research work on the field of perception for intelligent vehicles is not new, the recent affordability of sensors has increased the development and implementation of state-of-the art approaches. In this dissertation, we have proposed a multi-sensor fusion approach using affordable sensors. The sensor configuration we employed to test the real application can be considered as a real hardware configuration present in a near-future automotive vehicles. We have used the advantages of lidar, radar, and camera to overcome their individual drawbacks. Unfortunately, whilst our results are promising, there are situations where the limitations of these three sensors can be coped with the use of more accurate technologies.

7.2.1 Short-term perspectives

The classification process plays a key role in our multi-sensor fusion architecture. The extraction of classification information is made using all the available data gathered from each sensor. Although our camera-based classifiers cover a broad set of objects of interest, we believe that including a training dataset with a higher variety of positive and negative profile samples can improve the overall performance of the camera-based classifiers. This extension can help to overcome seldom situations such as when a traffic post is mis-classified as a pedestrian. Another possible solution to this issue is to include a specific classifier for static objects of interest, e.g, traffic-related obstacles, which moreover would enrich the final output of our perception solution.

The areas of application of perception for intelligent vehicles are mainly focused on outdoor environments, specifically highly dynamic driving scenarios. However, the core of the perception approaches we have presented in this dissertation can be ported to the problem of perception for indoor environments. For example, the SLAM and DATMO solutions can be modified to cover a different set of objects of interest, such as persons or indoor landmarks. Moreover, a further extension for our solution can be made by integrating range cameras to provide visual and three-dimensional views

of the environment. Perception for indoor environments is used as the first stage of service robotics and surveillance applications.

7.2.2 Long-term perspectives

The use of a 3D laser scanner as a reference sensor is an important extension of the work presented in this dissertation. We identified scenarios where the false classification objects could be avoided by using the three-dimensional shape of the object as a discriminating factor. A 3D representation can allow the ability to have more information about the shape of the obstacles surrounding the vehicle demonstrator. Moreover, a 3D segmentation process can be used to identify the moving objects of interest. Using 3D segments as Regions of Interest, a point cloud classification can be implemented to extract class information. Doing this could allow not only to focus on the objects of interest, but in the common obstacles that usually generate false detections and false classifications, such as: trees, bushes, poles, and traffic signs.

The addition of 3D information from a lidar scanner can take the role of an extra source of evidence inside the multi-sensor fusion approaches presented in Chapter 4 and Chapter 5. A 3D sensor can provide not only more evidence of the objects appearance, but it can be used to construct a three-dimensional representation of the map by implementing SLAM. It is important to note that, whilst a 3D lidar scanner is a promising addition to a perception system, the state-of-the-art research for 3D map representation, object segmentation, and classification is still working on real-time extensions of the most relevant work proposals (Azim, 2013). Moreover, the affordability of this kind of sensor is still far from being considered as a commercial deliverable.

As was shown in the experiment sections from Chapters 4, 5 and 6, the classification precision in challenging situations some times depends on the driving scenario, e.g, the vehicle classification is more accurate in highways than in cluttered environments. Several research works on the field of scene classification have shown promising results that can allow a perception system to provide general information about the driving scenario the vehicle is immerse in (Gangqiang *et al.*, 2012, Kastner *et al.*, 2009, Oliva and Torralba, 2001). This contextual information can be used to learn parameters about the performance of the sensor detection and classification modules according to the driving scene, thus providing reliability factors closer to the real driving situation.

References

Adams J.L. *Remote Control with Long Transmission Delays*. PhD thesis, Stanford University, 1961.

Andriluka Mykhaylo, Roth Stefan, and Schiele Bernt. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE Conference Publications, June 2008. ISBN 978-1-4244-2242-5.

Azim Asma. *3D Perception of Outdoor and Dynamic Environment using Laser Scanner*. Phd thesis, University of Grenoble 1, 2013.

Azim Asma and Aycard Olivier. Detection, classification and tracking of moving objects in a 3D environment. In *2012 IEEE Intelligent Vehicles Symposium*, pages 802–807. IEEE Conference Publications, June 2012. ISBN 978-1-4673-2118-1.

Baig Qadeer. *Multisensor Data Fusion for Detection and Tracking of Moving Objects From a Dynamic Autonomous Vehicle*. PhD thesis, University of Grenoble1, 2012.

Baig Qadeer, Aycard Olivier, Vu Trung Dung, and Fraichard Thierry. Fusion between laser and stereo vision data for moving objects tracking in intersection like scenario. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 362–367. IEEE Computer Society, June 2011. ISBN 978-1-4577-0890-9.

Baltzakis Haris, Argyros Antonis, and Trahanias Panos. Fusion of laser and visual data for robot motion planning and collision avoidance. *Machine Vision and Applications*, 15(2):92–100, December 2003. ISSN 0932-8092.

Bar-Shalom Y and Li XR. *Estimation and tracking- Principles, techniques, and software*, volume 38. A Wiley-Interscience publication, 1998. ISBN 0890066434.

Bar-Shalom Yaakov and Tse Edison. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, 1975.

Bellet Aurelien, Habrard Amaury, and Sebban Marc. A Survey on Metric Learning for Feature Vectors and Structured Data. *CoRR*, abs/1306.6, 2013.

Bertozzi Massimo, Broggi Aalberto, and Castelluccio Stefano. A real-time oriented system for vehicle detection. *Journal of Systems Architecture*, 43(1-5):317–325, 1997.

Bertozzi Massimo, Broggi Alberto, and Fascioli Alessandra. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous systems*, 32:1–16, 2000.

Beymer David, Rd Harry, Jose San, Konolige Kurt, Ave Ravenswood, and Park Menlo. Tracking People from a Mobile Platform. In *Experimental Robotics VIII*, pages 234–244. Springer Berlin Heidelberg, 2003.

Blackman Samuel S. Multiple Hypothesis Tracking For Multiple Target Tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, 2004.

Blanc C, Trassoudaine L, and Moreira R. Track to track fusion method applied to road obstacle detection. In *Information Fusion, 2004 7th International Conference on*. IEEE Conference Publications, 2004.

Bosch Anna, Zisserman Andrew, and Munoz X. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.

Broggi A, Bertozzi M, Fascioli A, and Sechi M. Shape-based pedestrian detection. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, number Mi, pages 215–220. IEEE Conference Publications, 2000. ISBN 0780363639.

Buluswar Shashi D. and Draper Bruce a. Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 11(2):245–256, April 1998. ISSN 09521976.

Burlet Julien, Aycard Olivier, Spalanzani Anne, and Laugier Christian. Adaptive Interacting Multiple Models applied on pedestrian tracking in car parks. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 462–467. IEEE Computer Society, 2006.

Burlet Julien, Vu Trung Dung, and Aycard Olivier. Grid-based localization and online mapping with moving object detection and tracking. Technical Report August, INRIA, 2007.

Castellanos J.A., Neira J., and Tardos J.D. Multisensor fusion for simultaneous localization and map building. *Robotics and Automation, IEEE Transactions on*, 17(6):908–914, 2001.

Chan Ching-yao and Bu Fanping. Literature Review of Pedestrian Detection Techonologies and Sensor Survey. Technical report, Institute of Transportation Studies University of California at Berkeley, 2005.

Chavez-Garcia R Omar, Burlet Julien, Vu Trung-dung, and Aycard Olivier. Frontal Object Perception Using Radar and Mono-Vision. In *2012 IEEE Intelligent Vehicles Symposium (IV)*, pages 159–164. IEEE Conference Publications, 2012. ISBN 9781467321181.

Chavez-Garcia R.Omar, Vu Trung-Dung, Aycard Olivier, and Tango Fabio. Fusion framework for moving-object classification. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1159–1166, 2013.

Chavez-Garcia R.Omar, Vu Trung Dung, and Aycard Olivier. Fusion at detection level for frontal object perception. In *2014 IEEE Intelligent Vehicles Symposium (IV)*, page [Pending for publication], 2014.

Chong CY, Booz Allen, Garren D., and Grayson T.P. Ground target tracking-a historical perspective. In *Aerospace Conference Proceedings, 2000 IEEE*, pages 433–448, Big Sky, MT, 2000. IEEE Conference Publications. ISBN 0780358465.

Civera Javier, Davison Andrew J, and Montiel J M M. Interacting Multiple Model Monocular SLAM. In *In proceeding of: 2008 IEEE International Conference on Robotics and Automation, ICRA 2008*, Pasadena, California, USA, 2008. IEEE Computer Society.

Cole D.M. and Newman P.M. Using Laser Range Data for 3D SLAM in Outdoor Environments. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1556–1563. IEEE Conference Publications, 2006. ISBN 0-7803-9505-0.

Coradeschi S. and Saffiotti A. Symbiotic Robotic Systems: Humans, Robots, and Smart Environments. *Intelligent Systems, IEEE*, 21(3):82–84, 2006.

Cox I.J. and Hingorani S.L. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996. ISSN 01628828.

Cucchiara R and Piccardi M. Vehicle Detection under Day and Night Illumination. *ISCS-IIA*, page 1999, 1999.

Cuevas Erik, Zaldivar Daniel, and Rojas Raul. *Kalman filter for vision tracking*. Number August. Freie Univ., Fachbereich Mathematik und Informatik, 2005.

Dalal N. and Triggs B. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE Conference Publications, 2005. ISBN 0-7695-2372-2.

Dang Hongshe, Han C., and GRUYER D. Combining of IMM filtering and DS data association for multitarget tracking. In *Proceedings of the International Conference on Information Fusion*, pages 876–880. IEEE Computer Society, 2004.

Dempster Arthur P. A Generalization of Bayesian Inference. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 73–104. Springer Berlin Heidelberg, 2008.

Dickmanns E.D. *Dynamic vision for perception and control of motion*. Springer, 2007.

Dickmanns E.D., Behringer R., Dickmanns D., Hildebrandt T., Maurer M., Thomanek F., and Schiehlen J. The seeing passenger car'VaMoRs-P'. In *Intelligent Vehicles '94 Symposium, Proceedings of the*, pages 68–73. IEEE Conference Publications, 1994.

Dissanayake M.W.M.G., Newman P., Clark S., Durrant-Whyte H.F., and Csorba M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001. ISSN 1042296X.

Dollár Piotr, Wojek Christian, Schiele Bernt, and Perona Pietro. Pedestrian detection: an evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–61, April 2012. ISSN 1939-3539.

Dubois D and Prade H. *Théorie des possibilités: applications à la représentation des connaissances en informatique*. M{é}thode + programmes. Masson, 1985.

Elfes A. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.

Farmer Michael E, Hsu Rein-lien, and Jain Anil K. Interacting Multiple Model (IMM) Kalman Filters for Robust High Speed Human Motion Tracking. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, number Imm, pages 20–23. IEEE Computer Society, 2002.

Fayad Fadi and Cherfaoui Véronique. Detection and Recognition confidences update in a multi-sensor pedestrian tracking system. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 409–416, 2008.

Floudas N., Lytrivis P., Polychronopoulos A., and Amditis A. On the track-to-track association problem in road environments. In *Information Fusion, 2007 10th International Conference on*, pages 1–8, 2007.

Fortmann T., Bar-Shalom Y., and Scheffe M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, July 1983. ISSN 0364-9059.

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

Freund Yoav. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

Friedman Jeorme, Hastie Trevor, and Tibshirani Robert. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337–655, 2000.

Gangqiang Zhao, Xuhong Xiao, and Junsong Yuan. Fusion of Velodyne and camera data for scene parsing. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1172–1179, 2012.

Garcia R., Aycard O., Trung-Dung Vu, and Ahrholdt M. High Level Sensor Data Fusion for Automotive Applications using Occupancy Grids. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 530–535, 2008.

Gavrila DM, Giebel J, and Munder S. Vision-based pedestrian detection: The protector system. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 13–18. IEEE Conference Publications, 2004.

Gerhardus Diana. Robot-assisted surgery: the future is here. *Journal of healthcare management/American College of Healthcare Executives*, 48(4):242, 2003.

Gidel S, Blanc C, and Chateau T. Non-parametric laser and video data fusion: Application to pedestrian detection in urban environment. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 626–632, 2009. ISBN 9780982443804.

Grabe Baerbel, Ike Thorsten, and Hoetter Michael. Evaluation Method of Grid Based Representation from Sensor Data. *Review Literature And Arts Of The Americas*, pages 1245–1250, 2012.

Grotzinger John P., Crisp Joy, Vasavada Ashwin R., Anderson Robert C., Baker Charles J., Barry Robert, Blake David F., Conrad Pamela, Edgett Kenneth S., Ferdowski Bobak, Gellert Ralf, Gilbert John B., Golombek Matt, Gómez-Elvira Javier,

Hassler Donald M., Jandura Louise, Litvak Maxim, Mahaffy Paul, Maki Justin, Meyer Michael, Malin Michael C., Mitrofanov Igor, Simmonds John J., Vaniman David, Welch Richard V., and Wiens Roger C. *Mars Science Laboratory Mission and Science Investigation*, volume 170. July 2012.

Gruyer D. and Berge-Cherfaoui V. Matching and decision for vehicle tracking in road situation. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 1, pages 29–34. IEEE Computer Society, 1999. ISBN 0-7803-5184-3.

Gruyer Dominique and Berge-cherfaoui Vkronique. Multi-objects association in perception of dynamical situation. In *UAI'99 Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 255–262, 1999.

Guo Ronghua. Interacting Multiple Model Particle-type Filtering Approaches to Ground Target Tracking. *Journal of Computers*, 3(7):23–30, 2008.

Hähnel D., Burgard W., Wegbreit B., and Thrun S. Towards Lazy Data Association in SLAM. In *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, number Ml. Springer, 2003.

Hall DL and Llinas James. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1), 1997.

Julier SJ and Uhlmann JK. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

Kaempchen Nico. IMM vehicle tracking for traffic jam situations on highways. In *Proceedings of ISIF/IEEE 7th*, volume 1. IEEE Conference Publications, 2004.

KaewTraKulPong P and Bowden R. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pages 135–144. Springer US, 2002.

Kalman R E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(Series D):35–45, 1960.

Kastner R., Schneider F., Michalke T., Fritsch J., and Goerick C. Image-based classification of driving scenes by Hierarchical Principal Component Classification (HPCC). In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 341–346, 2009.

Khan Zia, Balch Tucker, and Dellaert Frank. An MCMC-based Particle Filter for Tracking Multiple Interacting Targets. In *Computer Vision - ECCV 2004*, pages 1–12, 2004.

Labayrade Raphaël, Gruyer Dominique, Royere Cyril, Perrollaz Mathias, and Aubert Didier. Obstacle Detection Based on Fusion Between Stereovision and 2D Laser Scanner. In *Mobile Robots: Perception & Navigation*, number February. Pro Literatur Verlag, 2007. ISBN 3866112831.

Leibe Bastian, Leonardis Ales, and Schiele Bernt. Combined object categorization and segmentation with an implicit shape model. *ECCV 04 Workshop on Statistical Learning in Computer Vision*, (May):1–16, 2004.

Leonard J.J. and Durrant-Whyte H.F. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, number 91, pages 1442–1447, 1991.

Li Dalong. Moving objects detection by block comparison. In *Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on*, pages 341–344, 2000.

Lu F and Milios E. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4:333—-349, 1997.

Lu Feng and Milios E.E. Robot pose estimation in unknown environments by matching 2D range scans. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 935–938. IEEE Conference Publications, 1994.

Maji S, Berg A C, and Malik J. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE Conference Publications, 2008.

Marchal P., Dehesa M., Gavrila D., Meinecke M., Skellern N., and Viciguerra R. eSafety-Final Report of the eSafety Working Group on Road Safety. Technical Report November, Information Soc. Technology Programme of the EU, 2005.

Mason M.T. and Salisbury J.K. Jr. *Robot hands and the mechanics of manipulation*. The MIT Press, 1985.

Mateo Lozano Oscar and Otsuka Kazuhiro. Real-time Visual Tracker by Stream Processing. *Journal of Signal Processing Systems*, 57(2):285–295, July 2008. ISSN 1939-8018.

Mercier David and Jolly Daniel. Object Association with Belief Functions , an application with vehicles. *Information Sciences*, 181(24):5485–5500, 2011.

Milch S and Behrens M. Pedestrian detection with radar and computer vision. In *Progress in Automobile Lighting*. IEEE Conference Publications, 2001.

Mitchel H.B. *Multi-Sensor Data Fusion: An introduction*. Springer, 2007.

Montemerlo Michael, Thrun Sebastian, and Dahlkamp Hendrik. Winning the DARPA Grand Challenge with an AI robot. In *In Proceedings of the AAAI National Conference on Artificial Intelligence*, number Gat 1998, pages 14–20, 2006.

Montesano L., Minguez J., and Montano L. Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-based Navigation. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 4556–4562. IEEE Conference Publications, 2005. ISBN 0-7803-8914-X.

Moras Julien, Cherfaoui Véronique, and Bonnifait Philippe. Credibilist Occupancy Grids for Vehicle Perception in Dynamic Environments. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 84–89, 2011a.

Moras Julien, Cherfaoui Véronique, and Bonnifait Philippe. Moving Objects Detection by Conflict Analysis in Evidential Grids. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, number Iv, pages 1120–1125. IEEE Computer Society, 2011b. ISBN 9781457708893.

Moravec Hans Peter. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford University, 1980.

Oliva A and Torralba Antonio. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Visionon*, 42(3):145–175, 2001. URL `http://link.springer.com/article/10.1023/A:1011139631724`.

Pagac D., Nebot E.M., and Durrant-Whyte H. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623–629, 1998. ISSN 1042296X.

Papageorgiou Constantine and Poggio Tomaso. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

Parodi P. and Piccioli G. A feature-based recognition scheme for traffic scenes. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, pages 229–234. IEEE Conference Publications, 1995.

Perrollaz Mathias, Roy Cyril, Hauti Nicolas, and Aubert Didier. Long Range Obstacle Detection Using Laser Scanner and Stereovision. In *ntelligent Vehicles Symposium, 2006 IEEE*, pages 182–187, 2006.

Pinto Nicolas, Barhomi Youssef, Cox D.D., and DiCarlo J.J. Comparing state-of-the-art visual features on invariant object recognition tasks. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 463–470. IEEE Computer Society, 2011.

Qiang Zhu, Yeh M.-C., Kwang-Ting Cheng, and Avidan S. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE Conference Publications, 2006. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.119.

Ramesh Jain, W.N. Martin, and J.K. Aggarwal. Segmentation through the detection of changes due to motion. *Computer Graphics and Image Processing*, 11(1):13–34, 1979.

Reid D. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979. ISSN 0018-9286.

Roberts Gareth. Markov chain concepts related to sampling algorithms. *Markov Chain Monte Carlo in Practice*, pages 45—–57, 1996.

Schmidt Jr and Rodney Albert. *A Study of the Real-time Control of a Computer-driven Vehicle*. PhD thesis, Stanford University, 1971.

Shafer Glenn. *A mathematical theory of evidence*. Princeton University Press, 1976.

Singer R.A. and Stein J.J. An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems. In *Decision and Control, 1971 IEEE Conference on*, pages 171–175. IEEE Conference Publications, 1971.

Singh Vivek Kumar, Wu Bo, and Nevatia Ramakant. Pedestrian Tracking by Associating Tracklets using Detection Residuals. *2008 IEEE Workshop on Motion and video Computing*, (c):1–8, January 2008.

Skutek M., Linzmeier D.T., Appenrodt N., and Wanielik G. A precrash system based on sensor data fusion of laser scanner and short range radars. In *Information Fusion, 2005 8th International Conference on*, page 8. IEEE Conference Publications, 2005. ISBN 0-7803-9286-8.

Smets P. The combination of evidence in the transferable belief model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):447–458, 1990.

Smets Philippe. The Transferable Belief Model for Belief Representation. 6156(Drums Ii):1–24, 1999.

Smets Philippe. Data fusion in the transferable belief model. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, volume 1, pages 21–33, 2000.

Smith Randall, Self Matthew, and Cheeseman Peter. Estimating uncertain spatial relationships in robotics. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, page 850, 1987.

Srinivasa N. Vision-based vehicle detection and tracking method for forward collision warning in automobiles. In *Intelligent Vehicle Symposium, 2002. IEEE*, pages 626–631. IEEE Conference Publications, 2002.

Stauffer Chris and Grimson W.E.L. Adaptive Background Mixture Models for Real-Time Tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 22–46, 1999.

Subramanian V, Burks T F, and Dixon W E. Sensor fusion using fuzzy logic enhanced kalman filter for autonomous vehicle guidance in citrus groves. *Transactions of the ASAE*, 52(5):1411–1422, 2009.

Sun Z., Bebis G., and Miller R. On-road vehicle detection using optical sensors: a review. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 585–590. IEEE Computer Society, 2004. ISBN 0-7803-8500-4.

Tango Fabio. Advanced multiple objects tracking by fusing radar and image sensor data. In *Information Fusion, 2008 11th International Conference on*, pages 1–7, 2008.

Thrun Sebastian. Probabilistic Algorithms in Robotics. *AI Magazine*, 21(April):92–109, 2000.

Thrun Sebastian, Burgard Wolfram, and Fox Dieter. *Probabilistic robotics*. The MIT Press, 2005.

Tomasi Carlo and Kanade Takeo. Detection and tracking of point features. Technical Report April, School of Computer Science, Carnegie Mellon University, 1991.

Tzomakas Christos and Seelen Werner Von. Vehicle detection in traffic scenes using shadows. Technical Report August, IR-INI, INSTITUT FUR NUEROINFORMATIK, RUHR-UNIVERSITAT, 1998.

Viola P. and Jones M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518, 2001a.

Viola Paul and Jones Michael. Robust real-time object detection. Technical report, Cambridge Research Laboratory, Cambridge, Massachusetts USA, 2001b.

Vu Trung-Dung. *Vehicle Perception : Localization , Mapping with Detection , Classification and Tracking of Moving Objects*. Ph.d. thesis, University of Grenoble 1, 2009.

Vu Trung-Dung and Aycard Olivier. Laser-based detection and tracking moving objects using data-driven Markov chain Monte Carlo. *2009 IEEE International Conference on Robotics and Automation*, pages 3800–3806, May 2009.

Wang CC and Thorpe Chuck. Simultaneous localization and mapping with detection and tracking of moving objects. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 842—-849. IEEE Conference Publications, 2002.

Wang C.C., Thorpe C., Thrun S., Hebert M., and Durrant-Whyte H. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007.

Welch Greg and Bishop Gary. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill Chapel Hill, NC, USA, 1995.

Wu Bo and Nevatia Ram. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *International Journal of Computer Vision*, 75(2):247–266, January 2007. ISSN 0920-5691.

Yager Ronald R. O the Relationship of Methods og Aggregating Evidence in Expert Systems. *Cybernetics and Systems*, 16(1):1–21, 1985.

Yilmaz Alper, Javed Omar, and Shah Mubarak. Object Tracking: A Survey. *ACM Computing Surveys*, 38(4):1–35, December 2006. ISSN 03600300.

Zhao Liang and Thorpe Chuck. Stereo and Neural Netwrok-Based Pedestrian Detection. In *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, volume 1, pages 298–303. IEEE Conference Publications, 1999.

Zhu Song-Chun, Zhang Rong, and Tu Zhuowen. Integrating bottom-up/top-down for object recognition by data driven Markov chain Monte Carlo. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, pages 738–745, 2000.

Zimmermann H. J. Fuzzy set theory. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(3):317–332, May 2010. ISSN 19395108.