UAH

Driver Behavior Evaluation by using Smartphones

Máster Universitario en Sistemas Electrónicos Avanzados
Sistemas Inteligentes
Departamento de Electrónica

Presentado por:
Eduardo Romera Carmena

Dirigido por:
Luis Miguel Bergasa Pascual

Alcalá de Henares, a 30 de junio de 2015

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Departamento de Electrónica

Máster Universitario en Sistemas Electrónicos Avanzados.
Sistemas Inteligentes

Trabajo Fin de Máster

"Driver Behavior Evaluation by using Smartphones"

Autor

Eduardo Romera Carmena

Director

Dr. D. Luis Miguel Bergasa Pascual

2015

Agradecimientos

Como un año se acaba quedando corto y el año pasado me encontraba raptado por los alemanes, aprovecho esta página para agradecer no sólo el apoyo recibido en la realización de mi TFM este último año sino como valoración de toda mi etapa académica en la universidad.

En ella he conocido (por suerte en el primer año) a la persona más especial de mi vida y de la que estoy profundamente orgulloso porque se acaba de licenciar en Medicina convirtiéndose en la Dra. Vida (hasta el apellido parece irle acorde). Sin su compañía, confianza y apoyo, no sólo esto no hubiera sido posible sino que se habría convertido en un tortuoso camino que prefiero no conocer.

Por supuesto siempre estaré agradecido por todo a mi familia, sobre todo a mis padres, sin los cuales no sería quien soy y sin su apoyo, motivación y gran reconocimiento del esfuerzo no habría llegado nunca hasta aquí. También a mis hermanos, y especialmente a mi abuela Josefa, cuya capacidad para seguir riéndose de la vida todavía a sus 94 años me sigue teniendo asombrado.

A mis amigos y compañeros de fatigas, Iván y Sergio, sin los cuales la carrera habría sido una aburrida tortura similar a si me hubiera sacado Filología Hispánica en vez de Teleco.

Y finalmente, y ya centrados en este último año y en los que quedan por venir, a la gente del grupo RobeSafe. Al que tengo que agradecer por casi todo es a mi tutor Luis Miguel Bergasa, no sólo por ser un crack como investigador y como gestor de nuestra motivación para la investigación, sino también por todas las oportunidades y apoyo que me ha dado y sobre todo por su gran calidad como persona. Y por supuesto también a mis compañeros y amigos del laboratorio Sergio (sí, el mismo de antes :-)) y Roberto. No sólo por sus contribuciones al trabajo sino por su ayuda constante en todo y sobre todo por lo agradable que se vuelve trabajar en el laboratorio junto a ellos. Por su buena compañía les perdono el que siempre me tengan el aire acondicionado en modo glaciar e incluso empiezo a acostumbrarme gratamente a esto de trabajar en el Polo Norte :P. Es broma, no se está tan mal...

Resumen

En la última decada, ha habido una creciente tendencia en desarrollar sistemas que ayuden a hacer la conducción más segura. Aunque estos sistemas se van incluyendo poco a poco en los nuevos vehículos comerciales, la mayoría tienen un coste alto, relegando estos beneficios a vehículos premium. Por otro lado, la expansión de los móviles como plataforma de sensado ha facilitado la captura y procesado asequibles de datos, contribuyendo notablemente a expandir la ciencia del "Data Analytics", que proporciona información muy rica a instituciones y empresas en cuanto a la toma de decisiones. En el caso concreto de los vehículos, los datos relativos a análisis de conducción son de gran interés para gobiernos y aseguradoras, que intentan introducir el concepto de "Pay-As-You-Drive" como forma de beneficiar a los conductores que tengan una conducción más eficiente y segura. En este contexto nació DriveSafe App, una aplicación que evalúa la conducción detectando y puntuando comportamientos de inatención, dando el feedback correspondiente al conductor. Este trabajo toma como punto de partida la aplicación original, en desarrollo desde hace dos años, para realizar mejoras que expandan sus capacidades como plataforma de análisis de la conducción. En primer lugar, se desarrolla un servidor, con el correspondiente módulo cliente en la aplicación, para gestionar los datos de conducción de todos los usuarios de una manera unificada, necesario dado que DriveSafe esta enfocada al uso masivo. En segundo lugar, se desarrolla un módulo de detección y seguimiento de vehículos frontales por visión con el fin de obtener más información del entorno dinámico durante la conducción. Este módulo se presenta como una contribución independiente en el campo de visión, que mediante técnicas de multi-escalado consigue reducir costes computacionales para hacer viable una implementación en "smartphones", manteniendo resultados de detección similares al estado del arte. Esta información enriquece el análisis de datos de Drivesafe frente a otros trabajos relacionados, que no incorporan la detección de vehículos frontales en su análisis debido a su complejidad y necesidad de sensores costosos (e.g. RADAR, LiDAR) o costes de procesado altos en el caso de usar visión computacional. En último lugar, se reorganiza y re-analiza la arquitectura completa de análisis de conducción de DriveSafe para añadir nuevos indicadores, proporcionados por la detección de vehículos y por la introducción de capacidades comunicativas con APIs online de información de carreteras. De esta forma se consigue expandir el análisis de DriveSafe a un amplio rango de variables independientes, de las que se extraen indicadores para proveer un gran conjunto de puntuaciones de conducción al usuario y ampliar el modelado de comportamiento incluyendo una clasficación en tres niveles: conducción normal, agresiva o somnolienta junto a la detección de eventos distractores.

Abstract

In the last decade, there has an been active research toward developing systems that make driving safer. Although these systems are being slowly introduced in the new commercial vehicles, the high associated cost relieves their benefits to premium vehicles. On the other hand, the advent of mobile sensing platforms facilitates the cost-effective capturing and processing of data, contributing remarkably to expand the science of "Data Analytics", which provides rich information to institutions and companies for decision making. On the specific case of vehicles, the data relative to driving analysis is of high interest to governments and insurance companies, that try to introduce the concept of "Pay-As-You-Drive" as a way of awarding drivers that keep an efficient and safe way of driving. In this context was born DriveSafe App, an aplication that evaluates driving by detecting and scoring inattentive behaviours, giving the corresponding feedback to drivers. This work takes as a starting point the original application, that has been in development for two years, to perform improvements that expand its capabilities as a platform for driving analysis. Firstly, a server is developed, with the corresponding client module in the application, in order to manage the driving data from all the users in an unified way, a necessity due to the aim of DriveSafe to the massive use. Secondly, a module for vision-based ahead vehicle detection and tracking is developed and integrated in the application, with the aim of obtaining more information about the dynamical environment during driving. This module is presented as a single contribution in the vision field, which by applying multi-scale techniques achieves to reduce computational costs and make viable an implementation for smartphones, keeping detection results that are similar to the state of the art. This information enriches the data analysis of DriveSafe against other related works, that do not include ahead vehicle detection in their analysis due to its complexity and necessity of expensive sensors (e.g. RADAR, LiDAR) or high associated processing costs in the case of computational vision. Finally, the complete architecture for driving analysis of DriveSafe is rearranged and re-analysed to add new indicators, provided by the vehicle detection and the introduction of communication capabilities with road information online APIs. This achieves expanding the analysis of DriveSafe to a broad range of independent variables, from which indicators are extracted to provide the user a wide group of driving scores and augment the behaviour modelling including a three-level classification: normal driving, aggressive or drowsy, together with the detection of distracting events.

Contents

1. Introduction **1**
1.1. Motivation . . . 1
1.2. Related work . . . 2
1.3. Objectives . . . 4
1.4. Document structure . . . 5

2. DriveSafe App **7**
2.1. Architecture . . . 7
2.2. Interface . . . 8
2.2.1. Start and calibration . . . 8
2.2.2. Driving Interface . . . 9
2.2.3. End of route . . . 9
2.3. Inertial sensors and GPS . . . 11
2.4. Lane detection . . . 11
2.4.1. Pre-processing . . . 11
2.4.2. Lane markings detection . . . 12
2.4.3. Lane tracking . . . 13
2.4.4. Lane drifting . . . 14
2.4.5. Lane weaving . . . 14

3. Data Server for DriveSafe **15**
3.1. Data capturing . . . 15
3.2. Client . . . 15
3.3. Server . . . 16
3.3.1. Server framework . . . 16
3.3.2. PHP code . . . 17
3.3.3. Data structure . . . 17
3.4. Data analysis . . . 17

4. Ahead Vehicle Detection and Tracking **19**
4.1. Related work . . . 19
4.2. System design . . . 20
4.2.1. Detector 1: Near window . . . 21
4.2.2. Detector 2: Far window (zoom) . . . 22
4.2.3. Detector 3: Intermediate window . . . 23
4.2.4. Data association . . . 23
4.2.5. Optical Flow . . . 23
4.2.6. Extended Kalman Filter . . . 23
4.2.7. Lane detection . . . 25
4.3. Results . . . 25

4.3.1. Detection performance 26
4.3.2. Comparison with related works 26
4.3.3. Computing performance 26
4.3.4. Detection windows comparison 28
4.4. Conclusion 29

5. Driver Analysis **31**
5.1. Indicators 31
5.1.1. Inertial sensors 31
5.1.2. GPS 33
5.1.3. Lane detection 34
5.1.3.1. Lane drifting 34
5.1.3.2. Lane weaving 34
5.1.4. Ahead vehicle detection 36
5.1.5. OpenStreetMap 37
5.2. Driver scoring 38
5.2.1. Sudden accelerations, brakings and turnings 38
5.2.2. Lane drifting 39
5.2.3. Lane weaving 39
5.2.4. Overspeeding 40
5.2.5. Car-following 40
5.2.6. Global score computation 41
5.3. Behaviour modeling 41
5.3.1. Drowsiness 41
5.3.2. Aggressiveness 42
5.3.3. Normal driving 43
5.3.4. Distractions 43
5.4. Score GUI 43
5.5. Experiments 44
5.5.1. Definition of test bed 45
5.5.2. Evaluation of event/indicator detection and associated score 45
5.5.3. Evaluation of driver profiling 47
5.5.4. Qualitative evaluation 50

6. Conclusions **53**
6.1. Main contributions 54
6.2. Future work 54

Bibliography **57**

List of figures

1.1. DriveSafe App running on iPhone 5. 2
1.2. DriveSafe's driving interface. 2
1.3. SignalGuru uses visual data to tell drivers the optimal speed. 3
1.4. Publicitary image of iOnRoad application. 3
1.5. Publicitary images AXA Drive. 4

2.1. Architecture of DriveSafe App. 7
2.2. Example captures of the start and calibration process. 8
2.3. Interface with augmented Reality. 9
2.4. Interface without augmented Reality. 9
2.5. Route selection. 10
2.6. Scores and route information. 10
2.7. Remarkable information of the route depicted on the map. 10
2.8. Example of videos recorded during a sudden turn and a sudden brake. 10
2.9. Initial phone positioning. 11
2.10. Coordinate frame of iPhone vs vehicle. 11
2.11. Illustrative example of lane detection process at night 12
2.12. Definition of lane parameters. 13

3.1. Diagram of LAMP software bundle. 16
3.2. Distribution of data in the server. 17
3.3. Representation of vehicle communication. 18

4.1. DriveSafe App running in a real environment. 20
4.2. Full system diagram of the vehicle detection and tracking module. 21
4.3. Example of the three vehicle detection stages working. 22
4.4. Pinhole Camera Model. 24
4.5. Representative images of detection on each of the dataset's subsets 25
4.6. Detection algorithm evaluated on the "Daylight" subset of the TME Dataset. 27
4.7. Detection algorithm evaluated on the "Sunset" subset of the TME Motorway Dataset. . . . 27
4.8. Comparative charts of our approach and other state of the art results. 28
4.9. Comparison of detections covered by each detection window. 29

5.1. Coordinate frame of iPhone vs vehicle. 32
5.2. Example of lane drifting. 34
5.3. Process of a changing a lane. 35
5.4. Examples of vehicle detection. 36
5.5. OpenStreetMap example of road information. 37
5.6. Examples of overspeed detection and estimation of current lane. 38
5.7. Examples of scores shown in the new interface 44

5.8. Examples of the map interfaces that collect some events produced during one of the test routes. 50

List of tables

4.1. Average computation time for detection modules. 29

5.1. Distraction event thresholds. 32
5.2. List of drivers and vehicles that performed the tests. 45
5.3. Results of event detection. 46
5.4. Results of Behaviour-simulated routes for Driver 1 (Audi Q5). 48
5.5. Results of Behaviour-simulated routes for Driver 2 (Subaru Legacy - 1993). 48
5.6. Results of Behaviour-simulated routes for Driver 3 (Citröen C4 -2006). 48
5.7. Results of Behaviour-simulated routes for Driver 4 (Citröen C4 -2014). 48
5.8. Complete list of DriveSafe scores per behaviour state and driver. 49

Chapter 1

Introduction

1.1. Motivation

In 2014, more than $25,700$ people died on the roads of the European Union [1]. Studies about accident causation, like NHTSA's [2], attribute 94 % of accidents to driver-related reasons, such as distraction or inattention. Therefore, research and development of prevention measurements focused on drivers is essential for reducing fatality on the roads.

In the last years, there has been active research toward developing systems that make driving safer [3, 4]. These include collision-avoidance, lane departure warning, blind spot warning and driver inattention monitoring systems. Some systems even trigger automatic steering when the vehicle drifts into another lane or brake before getting dangerously close to the vehicle in front. While these systems are quite valuable in enhancing the safety, they are pricey too. Therefore, these safety features are commonly fitted only in top-end vehicles. Toward developing an affordable alternative for bringing safety features to economy vehicles our approach is to leverage smartphones that are always present with people.

On the other hand, the advent of mobile sensing platforms facilitates the cost-effective capturing and processing of data from the real world, thus increasing the information base of business processes and decision making [5]. In the motor insurance sector, such data can be used to improve the assessment, communication and mitigation of insured risk, thereby creating value for insurers and policyholders alike. The premise of this approach is that providing feedback of recorded driving actions to drivers, they are encouraged to change their behavior and reduce their individual accident risk. However, in the current insurance markets, consumers have rejected the so-called Pay-As-You-Drive due to two main reasons [6]: the required installation of "black-boxes" in vehicles makes drivers perceive the monitoring as intrusive, and the installation and operation of these units incurs additional costs to insurers and consumers. Our alternative approach is to use a smartphone application that is operated at the users discretion, emphasizing that it is more a driving support tool than a "black-box" monitoring device.

With this background, RobeSafe group proposed and presented DriveSafe App [7]. The driver's iPhone must be placed on the wind-shield, just below the rear-view mirror and aligned with the relevant axes of the vehicle, as it is depicted in Fig.1.1. Using the information obtained from some inbuilt iPhone sensors, DriveSafe applies computer vision and pattern recognition techniques on the phone to detect the most commonly occurring inattentive driving behaviours divided into two main groups: drowsiness and distractions. Lane weaving and drifting behaviours are measured to infer drowsiness. Lane weaving happens when a driver performs lane changes without turning the blinkers. Lane drifting is the inability of the driver to keep its vehicle within the centre of the lane. Distractions are based on sudden longitudinal and transversal movements. In addition, the app scores the driving as a function of the frequency and level of these dangerous behaviors. In case they get over a certain threshold an alarm is generated. On the

Figure 1.1: DriveSafe App running on iPhone 5.

Figure 1.2: DriveSafe's driving interface.

one hand, DriveSafe aims to mimic some safety features found in many top-end vehicles on the market today but using a commodity iPhone. On the other hand, it persuades insurers this app is an interesting alternative to conventional “black-boxes”, improving other proposals of the state of the art.

The present master thesis takes as a starting point the application DriveSafe presented in [7] to try to enhance all its features. Three main emphases have been followed in the development of this thesis as the targets to improve in the original version of DriveSafe. Firtly, we focused on the fact that the application lacked communication capabilities. As a platform that is intended for massive use, the right structuring of the data captured by the drivers and the separation between the application of the common user and the data managed by our laboratory is a necessity and is achieved by the development of a client/server that manages the upload of driving data for posterior analysis. Secondly, vehicles in the road give rich information about driving behaviour. This is something that is not treated in most of the works related to driver analysis due to the complexity of a vehicle detection system, which may be either performed with complex sensors (RADAR or LiDAR) or with vision (with high associated processing costs). In this work, we present a vision-based ahead vehicle detection system that is expandable to any smartphone and produces a rich source of information in the driver analysis, as it allows extracting indicators such as the distances kept to ahead vehicles, an estimation of the traffic or information about driver's overtakings. Finally, the complete driver scoring and profiling system of DriveSafe has been analysed and rearranged in order to cover several new indicators that are detected by DriveSafe, resulting in a richer analysis. A wide set of scores based on several indicators is introduced in DriveSafe, together with an augmented set of behaviour types classification (Aggressiveness, Drowsiness, Normal and specific Distraction events). All these contributions are presented in this thesis work both as an enhancement of the theoretical analysis that is the base to DriveSafe App, which is also evaluated experimentally in several tests, and as an expansion of the features and interfaces that are offered in the application to the final user.

1.2. Related work

Monitoring driving behavior using fixed vehicle-mounted devices is an active area of research. In the case of inattentive driving, a good review of the current state of the art can be found in [8]. The systems used by many auto manufacturers are mainly based on core technology from Mobileye [3] and Iteris [4]. Both companies use radars as well as cameras for this purpose. However, none of the cited examples consider the limitations and challenges of a smartphone-based implementation.

Although the cost of vehicle safety technology is dropping, most safety technologies are not available in economy vehicles and it will be a decade before the vast majority of cars on the road today have these safety features built-in. In contrast, smartphone solutions can be used in all vehicles (new or old) and represent a cheap and disruptive technology. This is the reason why in the last years there has been an

Figure 1.3: SignalGuru uses visual data to tell drivers the optimal speed.

active work on using smartphones to assist drivers. Hereafter, we review some of the most important references. SignalGuru [9] (Fig. 1.3) advises the driver to maintain a certain speed while approaching a signal for fuel efficiency. iOnRoad [10] (Fig. 1.4) and Augmented Driving [11] are apps that warn drivers when they get too close to a vehicle. SmartLDWS [12] offers warning sounds when the vehicle departs a lane marker. CarSafe [13] alerts drowsy and distracted drivers using dual cameras on smartphones, one for detecting driver state and the other for tracking road conditions. However, it works in quasi real-time and the driver's indicators get worse at night and with bad lighting conditions. Aviva RateMyDrive [14], StateFarm DriverFeedback [15] and AXA Drive [16] (Fig. 1.5) appear to be the most popular mobile applications to score driving for insurances companies purposes. Greenroad [17] is an online platform for fleet management. In [18], a recognition system of the driver aggressiveness based on sensor-fusion is presented. In the same line, [19] estimates if a driving behavior is safe or unsafe. The last two cases need a previous learning and are dependent of the road curvature. Other works have proposed to extend smartphone's sensors capabilities by connecting them to a vehicle's OBD-II diagnostic interface for driver classification [20] or reduction of fuel consumption purposes [21].

Figure 1.4: Publicitary image of iOnRoad application.

More recently, researchers in [22] proposed SenseFleet, which applies fuzzy logic to fuse inertial sensors together with road and weather information (accessed via online APIs) in order to score and profile driver on basis of overspeeding and acceleration events. Most of the above applications focused on driver behavior profiling are based on inertial sensors and GPS. Existing solutions for event detection are commonly based on fixed thresholds for the different input variables used to detect acceleration, braking or steering events. Finally, there are other interesting approaches that uses dedicated hardware or data taken from the CAN bus of the vehicle as [23] and [24].

Figure 1.5: Publicitary images AXA Drive.

In developing DriveSafe, some techniques available in RobeSafe's group were reviewed [25, 26] and adapted them to be run on an iPhone. To the best of our knowledge, none of the related works detects driver inattentions (drowsiness and distractions) by using lane weaving/drifting, sudden longitudinal/transversal movements from the inbuilt sensors of an iPhone, and GPS information, evaluating the quality of the driving at the same time, independently of the road geometry and running in real- time. DriveSafe was proposed to fill this gap. Additionally, this thesis introduces in DriveSafe App a module for vehicle detection and connection to online map APIs, expand driver analysis to a point that remains unexplored in the state of the art, due to the wide set of capabilities that DriveSafe seizes in the smartphone to obtain driving indicators and profile behaviours such as drowsiness, aggressiveness and normal.

1.3. Objectives

By the start of this thesis, DriveSafe App had already been on development for two consecutive years. Therefore, the purpose of this work is not to create from zero an application such as DriveSafe App, but to contribute and develop modules that enhance its functionality and try to transform it in one of the most complete sensing platforms that there is available for smartphones in the present time. Thus, the objectives of this work might be divided in three main contributions:

- **Data server.** A sensing platform that is designed for massive use requires a server that manages all the data captured by the users. Client and server sides have been designed to provide DriveSafe communication services to seize and analyse the captured information.
- **Ahead vehicle detection and tracking module.** Surrounding vehicles significantly affect driving and the distances kept to ahead vehicles (e.g. tailgating) may also give information about behaviours such as aggressiveness, drowsiness or distractions. With the aim of improving driver analysis, a full vehicle detection and tracking module has been developed and integrated in DriveSafe.

It is presented as a single contribution to the vision field that might be used in any smartphone or on-road vision system.

- **Driver analysis.** The new modules (vehicle detection) and additions such as a connection to road information APIs allows to introduce new indicators to the previously developed ones with the aim of enhancing driver analysis. The integration of these new variables produces a rearrangement of the driver scoring and behaviour profiling algorithms into a more complete proposal, which is now extended to an evaluation between driving states (normal, drowsy and aggressive) and specific driver distractions.

1.4. Document structure

Chapter 2: DriveSafe App
This chapter introduces DriveSafe. It describes the previously developed contents of the application, which is the starting point of the contributions of the present work.

Chapter 3: Data Server for DriveSafe
It describes the server and client sides of the application that manages the data captured by DriveSafe and uploaded to a server, for posterior analysis and feedback for debugging.

Chapter 4:Ahead Vehicle Detection and Tracking
This chapter describes the design and evaluation of the vehicle detection and tracking module implemented in DriveSafe. It contains its own Related work and Results sections, as it supposes a single contribution on the field of computer vision.

Chapter 5: Driver Analysis
It explains how different indicators can be extracted from the phone features and how these indicators are analysed to produce a series of driving scores and behaviour models that profile the driver. An evaluation is performed to test both the detection of driving indicators and the assignment of scores and behaviour ratios on different conditions.

Chapter 6: Conclusion
It contains conclusions and future work discussion.

Chapter 2

DriveSafe App

This chapter introduces DriveSafe App. DriveSafe started its development prior to the present work. This chapter describes all the features that had been already developed and constitute the starting point of this thesis.

DriveSafe is a sensing platform that collects data to evaluate and profile driver behaviours. During driving, it shows information of the road by augmented reality and alerts driver of several events. At the end of each trip, the driver receives a score and is allowed to review all the trip information to learn how they performed and what they can do to improve their driving skills. DriveSafe is strictly not designed to replace, substitute or complement any vehicle control system but to alert and evaluate driving behaviours to encourage safe driving.

2.1. Architecture

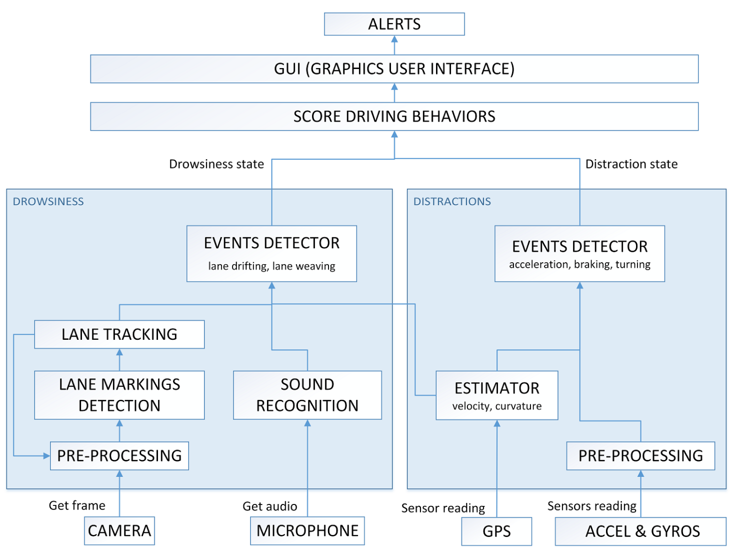


Figure 2.1: Architecture of DriveSafe App.

DriveSafe makes use of all the available sensors on the Smartphone to evaluate driving. Fig. 2.1 shows the full architecture of the application. The images captured by the camera are processed in order to detect the road markings and obtain an estimation of the vehicle position with respect to the driving lane. The microphone is used to analyse the turn signals to infer whether a lane change is voluntary or involuntary. The GPS provides speed and course information. Finally, the inertial sensors (accelerometers and gyroscopes) are used to detect sudden acceleration, braking and turning events. All together is used to produce scores and infer driving states (drowsiness and distraction) according to the lane weaving and drifting, and the inertial events.

The present thesis adds modules to the architecture that are not present in the scheme in Fig. 2.1. As will be explained in Chapter 3, a communication module is added to connect with a server specifically designed for management of data captured by DriveSafe. Chapter 4 will describe the vehicle detection and tracking module that is integrated in DriveSafe and detects ahead vehicles by means of the rear camera. Finally, Chapter 5 will describe the rearrangement of the new indicators, such as the vehicle detection and the access to road information via internet, to produce richer behaviour modelling based on 7 indicators.

2.2. Interface

DriveSafe App was developed with the aim to evaluate the drivers and give feedback in order to help them improve their driving. A clear interface achieves this target while avoiding to introduce additional distractions in the driving environment.

2.2.1. Start and calibration

As it is shown in Figs. 2.2, the initial interface allows the user to start a new route by introducing a reduced set of parameters of the vehicle and performing a simple calibration process which sets the smartphone in the correct axis with respect to the ground. This process is assisted by the smartphone gyroscopes, so the driver must only make sure that the phone holder is approximately centred in the wind-shield and then slightly rotate it until two simple axis lines painted on DriveSafe interface become aligned.

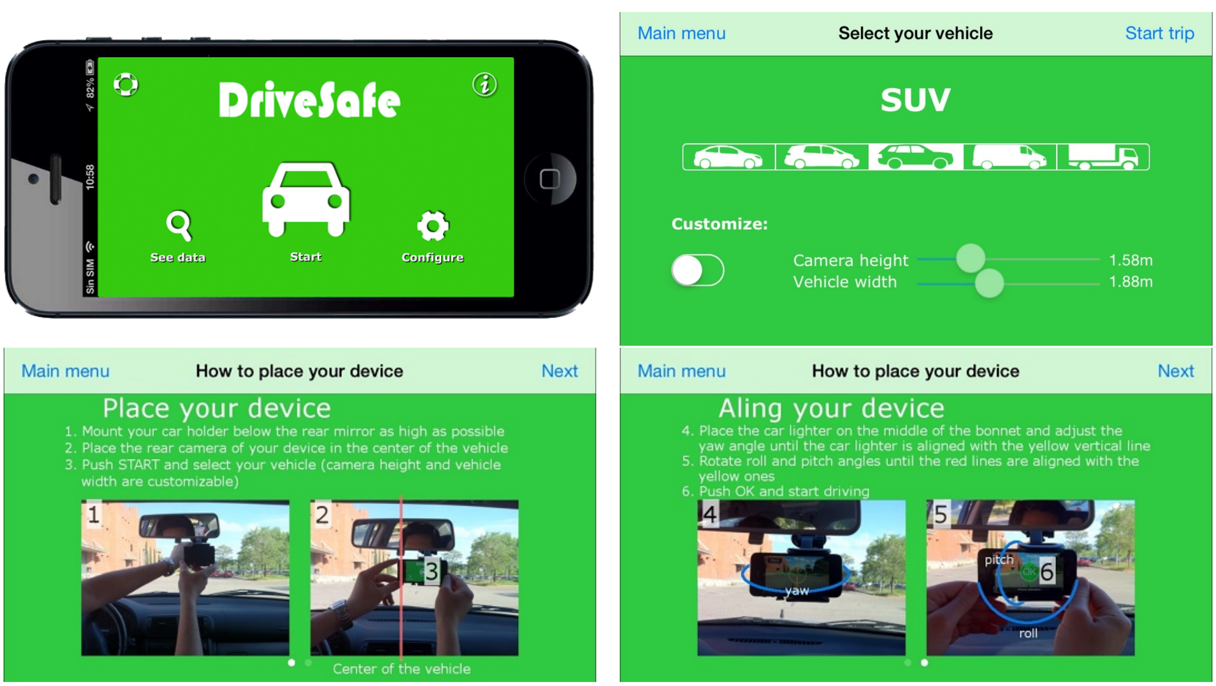


Figure 2.2: Example captures of the start and calibration process.

2.2.2. Driving Interface

During the driving, DriveSafe shows road information in the form of augmented reality, as shown in Fig. 2.3. This information comprises:

- Detected road lane, with a coloured indicator that depicts position inside lane.
- Detected vehicles, with their distance and time to impact (respect to own speed).
- Speed obtained by GPS.
- Inattention indicator with 6 coloured levels.
- Several labels and cartels that pop up on different events such as lane departures, rest recommendations, etc.

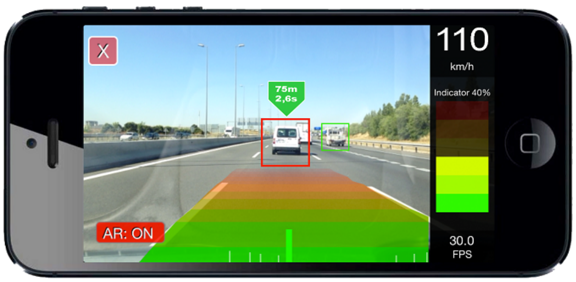


Figure 2.3: Interface with augmented Reality.

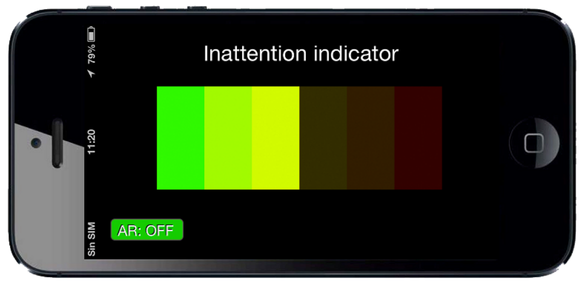


Figure 2.4: Interface without augmented Reality.

With the aim of avoiding the driver to regularly look at the smartphone and hence distracting the driver, DriveSafe also produces sounds to indicate some events, such as "too close to ahead vehicle", "close to lane change" and "lane change produced". The user might also disable the augmented reality interface to avoid any distraction. Fig. 2.4 shows this simplified interface, which depicts only the inattention level on the full screen.

2.2.3. End of route

A view in the form of a list of trips (Fig. 2.5) allows the user to review each of the trips performed with the application. By clicking individually on each item, another view (Fig. 2.6) shows all the information available for that specific route: the list of scores, the amount of inertial events, route statistics, access to map view and recorded videos.

The map view (Fig. 2.7) allows the user to review the trip route and find all the driving events located on the map. Different tags display the importance of each event and the route sections are coloured according to the inattention score that the driver had on that specific kilometre.

DriveSafe also records short videos on the detection of very remarkable inertial events. On a normal driving environment, events such as accelerations, brakes and turns are common during the driving operation and do not necessarily suppose a dangerous situation. However, a significant event such as a sudden brake that highly exceeds the average braking level might indeed precede an accident. Upon the detection of a sudden event, a video buffer of 10 seconds (kept during driving) is put together with the ten seconds posterior to the event, producing a video of a total duration of 20 seconds. The speed during the recording is tagged in each frame, as depicts Fig. 2.8. These videos might be reviewed together with the route information and they might be useful for the driver to learn what was done wrong to produce that event (e.g. following ahead vehicle too close) or even as a tool to clarify an accident.

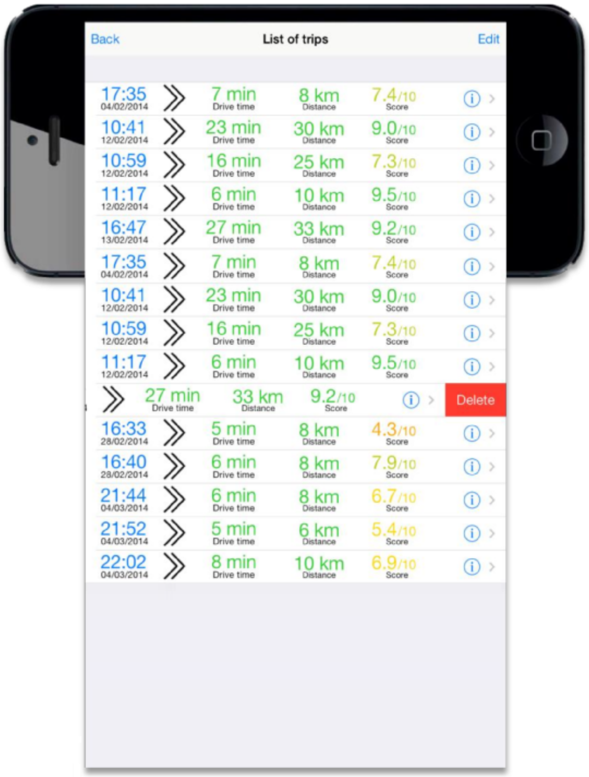


Figure 2.5: Route selection.

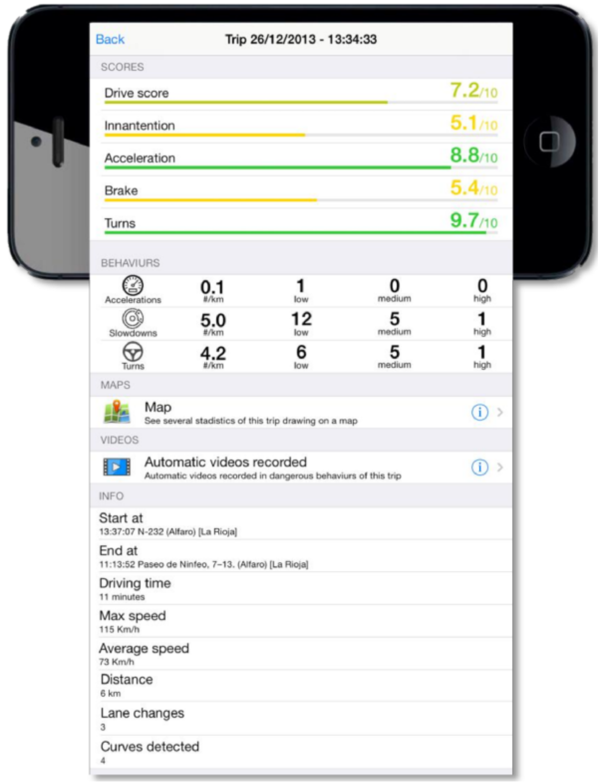


Figure 2.6: Scores and route information.

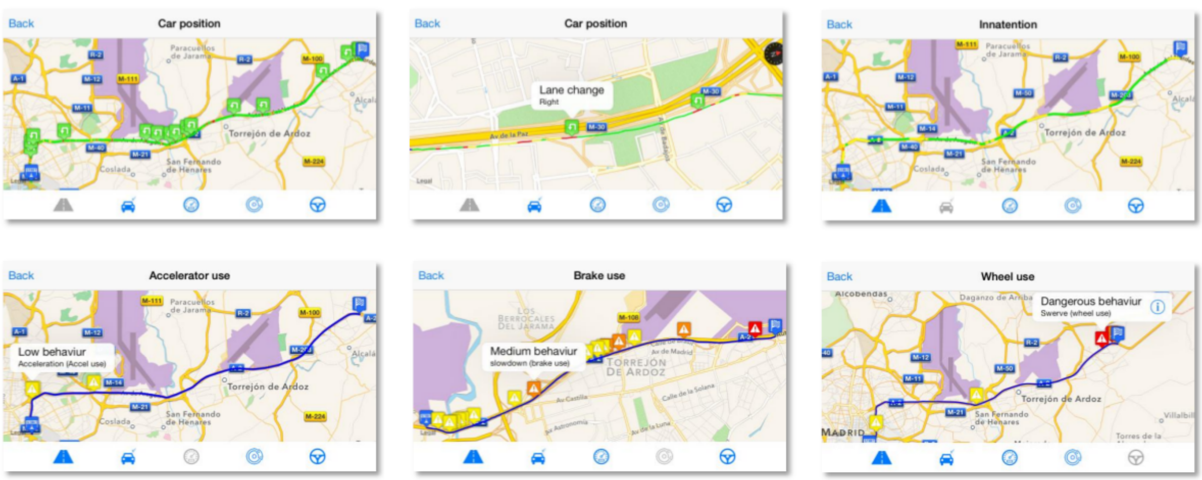


Figure 2.7: Remarkable information of the route depicted on the map.

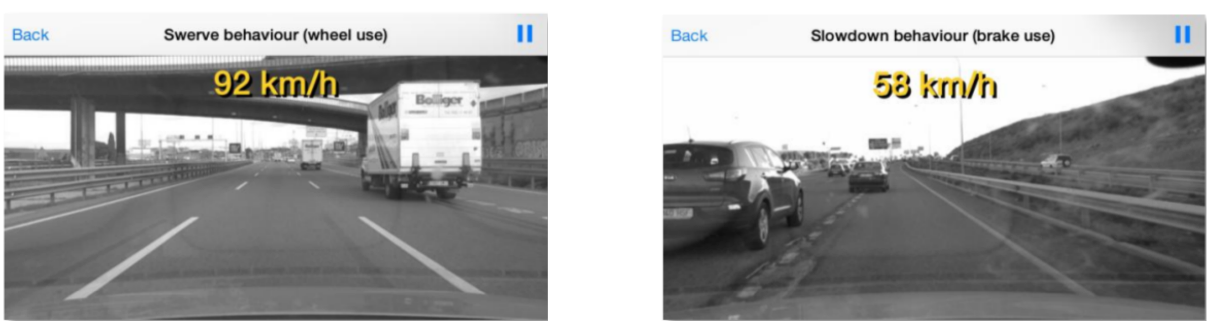


Figure 2.8: Example of videos recorded during a sudden turn and a sudden brake.

2.3. Inertial sensors and GPS

As shown in Figs. 2.9 and 2.10, the smartphone is positioned in a specific way on the calibration process. This allows to differentiate the driving axes between longitudinal and lateral. Thus, the accelerometer values in axis Z are used to detect sudden accelerations and brakings, and the values in axis Y are used to detect sudden turnings on the road. Sec. 5.1.1 will describe the thresholds that are set set to the accelerometer measurements in order to detect between these three types of events and three different levels for each one: low, medium and high.

Figure 2.9: Initial phone positioning.

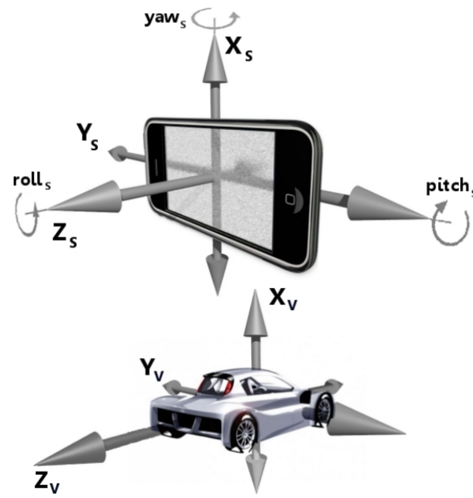


Figure 2.10: Coordinate frame of iPhone vs vehicle.

DriveSafe makes also use of the GPS available in the smartphone to obtain the vehicle speed each second. The heading/course value (i.e. the direction in which the device is traveling) is used to correct turning events that might be detected during a road curve and are not supposed to be driver's fault.

2.4. Lane detection

Lane detection is performed by a modification of the Dickmans road model-based method [27], which is fully explained in the work that presented DriveSafe App [7]. The combined effect of low resolution and lack of real camera parameters has made unreliable the deployment of some algorithms based on Inverse Perspective Mapping. It employs road images captured by the rear-camera to detect lane markings on the road ahead and to estimate the position of the vehicle relative to the lane and lane crossings. The GPS is used for estimating vehicle speed and road curvature. The lane detection algorithm is robust to failures and guaranties a correct detection. It is based on three functionalities which are contained in the pre-processing module: calibration, initialization and detection.

2.4.1. Pre-processing

This module transforms the color image to gray scale, resizes it to 320 x 240 pixels and adapts the ROI for the lane markings detection in the next frame. There is only a trapezoidal ROI in the calibration and initialization, but in the detection it is divided in two, one for the left markings and another for the right one (see Fig. 2.11a). The width of these areas is adaptive and depends on the tracking module. As long as the tracking gets worse, the width increases, and vice versa. Then, this module is in charge of the switching among the different functionalities based on tracking performance.

2.4.2. Lane markings detection

Lane markings are assumed to be edges. Then, an edge detection stage is carried out by using an adaptive Canny algorithm which maximizes the edges in each ROI (see Fig. 2.11b). After that, the Hough transform algorithm is applied to obtain candidate lines for each of the two ROIs, limiting the solution space by imposing some geometrical constraints (see Fig. 2.11c). Among all the segmented lines, we choose a representative line per ROI maximizing the length of the line, minimizing the angle difference between the candidate and the road model and the difference between the model vanishing point and the vanishing point obtained among the candidates for the left and the right side. After this process, we will have a winner line for each ROI aligned with the lane boundaries (see Fig. 2.11c winner lines in white color).

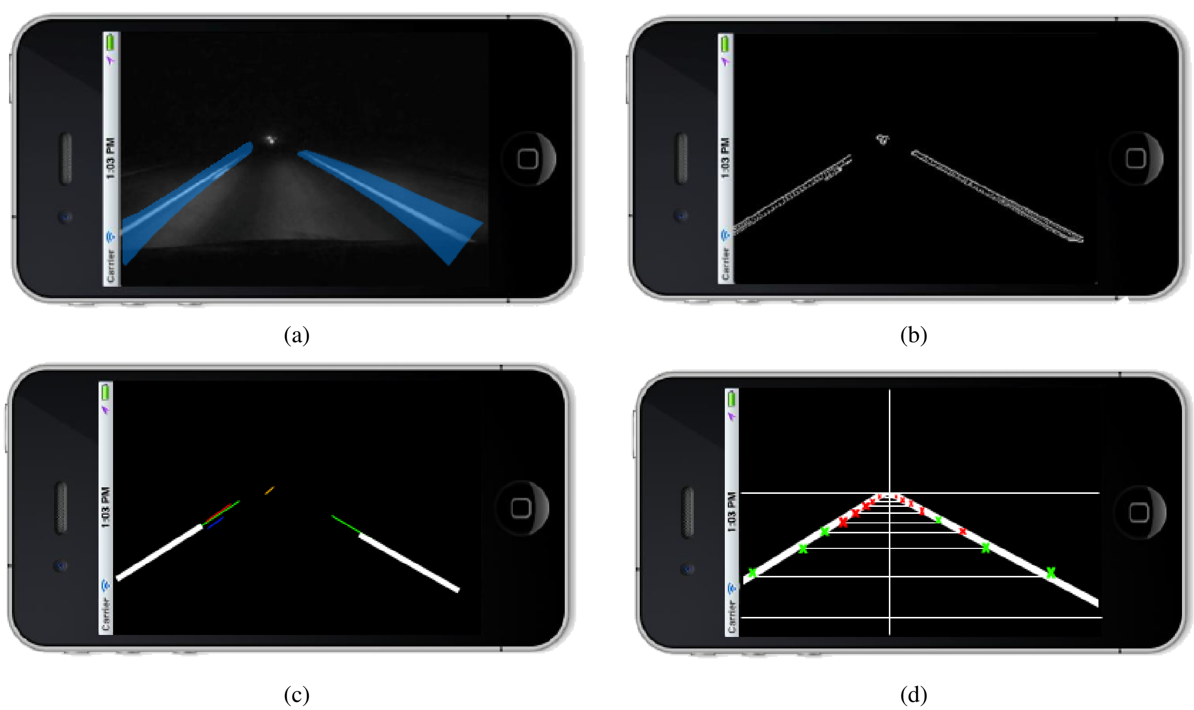
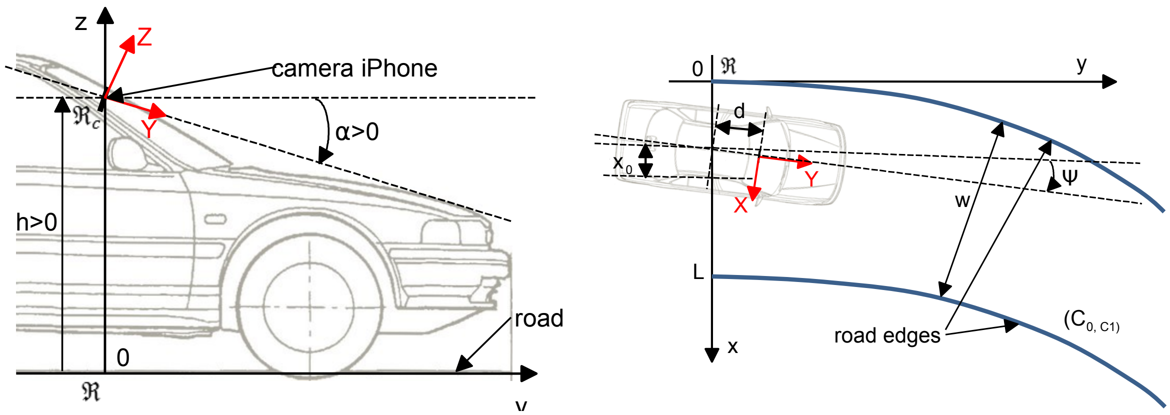


Figure 2.11: Illustrative example of lane detection process at night. a) ROIs in the gray scale image, b) Canny edges, c) Segmented and winner lines, d) Marker measures and lane model.

To increase detection robustness we use a 3D road model in the real world and the position of the road edges in the image, following a clothoidal model defined in [27]. The mathematical problem is that of computing the clothoidal parameters as well as road parameters and the position of the ego-vehicle with respect to the road edges. This amounts to a total of 5 different parameters: C_0 (initial curvature), C_1 (velocity of the clothoidal curve), w (road lane width), x_0 (lateral displacement of the car with regard to the center of the lane), and ψ (angular displacement of the car with regard to the lane orientation). The meaning of these parameters is described in Fig 2.12.

Measurements in the image plane must be related to measurements in the 3-D scene. Following the perspective projection equations of a camera and the equations of clothoidal curves, a measurement model is set as described in Eqs. 2.1 and 2.2.



(a) (b)

Figure 2.12: Definition of lane parameters. a) Side view. b) Bird's eye view.

$$\theta = a\tan\left(\frac{h}{L}\right) \tag{2.1}$$

$$v = f_v \cdot \tan\left(\theta - \alpha\right) \tag{2.2}$$

where v stands for the vertical coordinate of the road edge feature in the image plane, h is the camera height, α is the camera pitch angle, L is the distance from the car to the edge feature in the 3-D scene, and f_v is the vertical size of the camera focal length. Horizontal coordinates are computed from Eqs. 2.3 and 2.4.

$$L_{CV} = L + d \tag{2.3}$$

$$u = \frac{f_u}{L} \cdot \left(C_0 \cdot \frac{{L_{CV}}^2}{2} + C_1 \cdot \frac{{L_{CV}}^3}{6} - x_0 \pm 0{,}5 \cdot w - L_{CV} \cdot \psi\right) \tag{2.4}$$

where d stands for the distance between the vehicle gravity center and the camera position, f_u is the horizontal dimension of the camera focal length, and u is the horizontal coordinate of the edge feature pixel in the image plane. Camera parameters, h and d are set in the calibration. Eqs. 2.3 and 2.4 is computed for all lane markings detected in the neighborhood of the winner lines for every meter (Fig. 2.11d points in red and green). Then, the road model is estimated (in white).

2.4.3. Lane tracking

It is implemented using Kalman filtering based on the previous measurement model and the dynamic state model for the following state vector:

$$x = [C_0 \quad C_1 \quad x_0 \quad \psi \quad w]^T \tag{2.5}$$

When the lateral position (x_0) is about to leave the lane, a new lane model is generated by shifting the current model to the left or the right a current lane width. If enough lane marking measures are detected with the new model, a lane switch is carried out. Otherwise, the current lane model is kept. A lane change is detected when the lateral position overtakes half of the lane width (w/2).

2.4.4. Lane drifting

Lane detection is used to estimate drifting, based on the indicator called Lanex (fraction of Lane exits), which is a measure of driver's tendency to exit the lane [26]. It is defined as the fraction of a given time interval spent outside a virtual driving lane around the center of 1.2 m width. It is calculated by applying windowing techniques over the lateral position of the vehicle (x_0) during 60 s.

2.4.5. Lane weaving

The second measurement estimated by lane detection is weaving, which evaluates involuntary lane changes. Analyzing the presence or absence of the directional indicator, the event detector module can conclude whether a lane change is intentional or not. Initially, DriveSafe used the built-in microphone to capture the clicking sound generated by the indicator, in order to avoid external dependencies. The sound recognition was enabled when the vehicle is about to leave the lane. In this work, microphone is disabled and lane weaving is further developed, as it will be explained in Sec. 5.1.3.

Chapter 3

Data Server for DriveSafe

DriveSafe is a sensing platform that collects plenty of information from each route. One of the core advantages of smartphones against other on-road platforms are their communication capabilities. Access to the internet is present in any smartphone and it opens several data analytics possibilities. DriveSafe is aimed towards massive use, so it is necessary to develop an entity that captures the data to allow further analysis. With these considerations, a data server has been designed to collect the uploads performed from a HTTP client inside DriveSafe.

3.1. Data capturing

Each trip that is performed with DriveSafe, the app saves both raw sensor measurements (e.g. accelerometer values) and all the processed information (e.g. distance to ahead vehicle, scores obtained). This data is saved in text files and arranged in columns with time tags, allowing posterior analysis as a sequence of measurements. The data comprises:

- GPS values: speed, course, altitude, horizontal and vertical accuracy (1Hz).
- Accelerometers and Gyroscope values: X, Y, Z, Pitch, Roll, Yaw (100Hz downsampled to 10Hz).
- List of events and their location: lane departures, sudden brakings, etc.
- Lane values extracted and road clothoidal model (30Hz).
- Information from ahead vehicle detection: distance to ahead vehicle, time of impact, traffic density.
- Information captured by accessing online map APIs (e.g. overspeeding).
- All driver scores and behaviour modelling as will be described in Chapter 5.
- Extra debugging information: FPS, errors, etc.

3.2. Client

A communication module has been developed in DriveSafe App to be able to contact with the server. Once the user ends a route, the application wraps all the data in a “zip”, which is moved into a specific folder for files that await uploading. DriveSafe generates an Unique Device Identifier (UDID) for each user, which is embedded into the file name. The final name of the uploaded file is:

DriveSafe_User(UDID)_PhoneType_RouteDate(YYYYMMDDhhmmss).zip

The application checks connectivity with the server regularly. Whenever it is possible, all files awaiting upload are sent one by one by an HTTP PUT request to the server, which is described in Sec.3.3.

3.3. Server

A server must be able to manage upload requests sent by DriveSafe users. It is currently available at `http://drivesafe.uah.es`. Both the infraestructure followed to design the server and the organization that it follows to arrange the uploaded data are described below.

3.3.1. Server framework

The server has been developed following LAMP model (Linux, Apache, MySQL, PHP), represented in Fig. 3.1. This architecture comprises:

- **Linux** operative system, which allows an easy management of the network resources and file storage.
- **Apache** is a webserver that implements HTTP protocol, allowing the reception of DriveSafe data via PUT requests from the client side.
- **MySQL** allows easy management of a database. This feature is currently not used but it will be used in the future for a more complex management of user data.
- **PHP** is a programming language that allows providing some intelligence to the server. In our case, the code that manages uploads is programmed in PHP and described in Sec. 3.3.2.

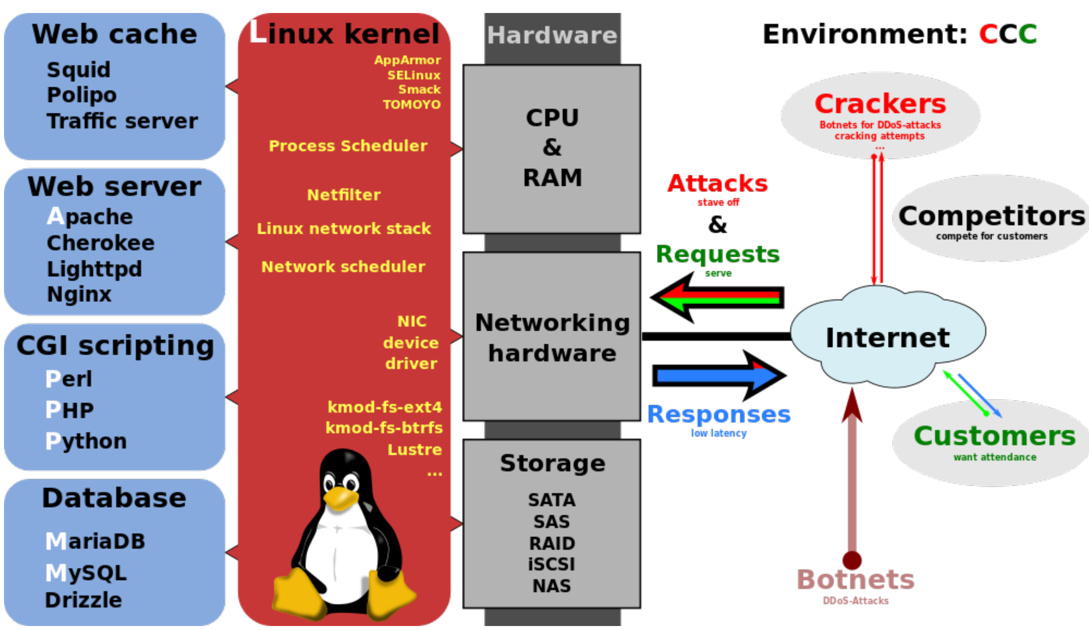


Figure 3.1: Diagram of LAMP software bundle.

3.3.2. PHP code

A PHP code manages PUT requests sent by the client side of DriveSafe. Each upload has to pass a set of filters to avoid data corruption. Firstly, delivery is protected by an user and password that is only available inside DriveSafe App's code, so upload is only allowed to the official application. Secondly, size and format of the uploaded files are checked to control that it follows the desired standards. Finally, the integrity of the file is checked and saved in the server file storage, following the structured that will be described in Sec. 3.3.3. The code also manages the creation of folders for each user or device when necessary (first-time uploads). If everything succeeds, the code sends back an HTTP response in the form of an ACK so the client is able to know that everything was received successfully.

3.3.3. Data structure

Fig. 3.2 depicts how the files are stored in the server by the PHP code. The name of the file sent by the client contains the identifiers that allow storage in different compartments. An application ID allows compatibility for using the server on other applications in the future. An user unique ID (UDID) is used to separate the data for each user. A phone ID splits the data according to the type of phone (e.g. iPhone 5, iPhone 6), as each one may have differences in indicators such as the accelerometers. Finally, each route performed by DriveSafe is saved in a different file which name is the exact date it was performed.

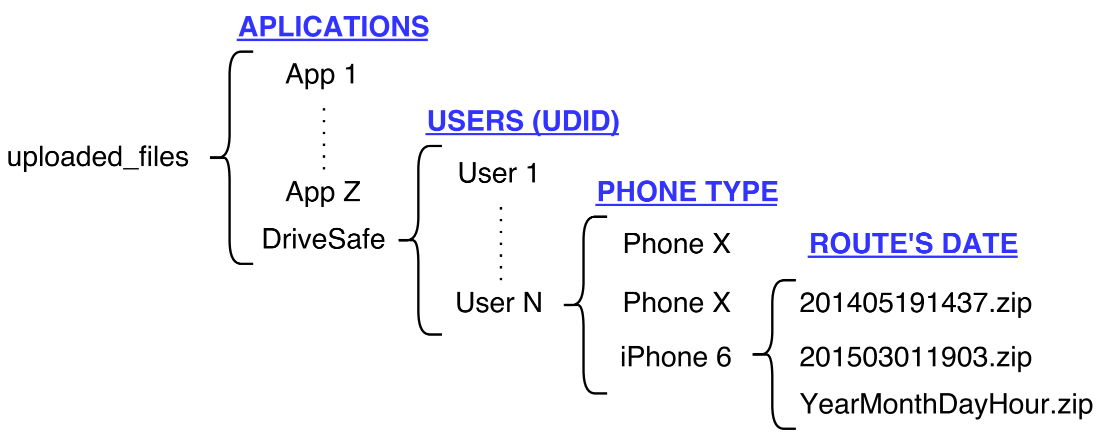


Figure 3.2: Distribution of data in the server.

Currently, data is stored as a zip. In future works, the PHP code may also be adapted to extract the files and perform further data analysis, as will be described in Sec. 3.4.

3.4. Data analysis

A server that collects the data from massive use is essential to cover several purposes:

- **Feedback for development**. Receiving data from each use has already proven useful for development of DriveSafe, and it becomes even more important in a massive use because it allows the correction of several software errors that remained hidden during debugging.
- **Hardware parameter adjustment.** Each smartphone has some differences on how their accelerometers capture movement or on the quality of the image retrieved by their cameras. Receiving data massively allows adjustments with respect to the device so DriveSafe may spread globally.

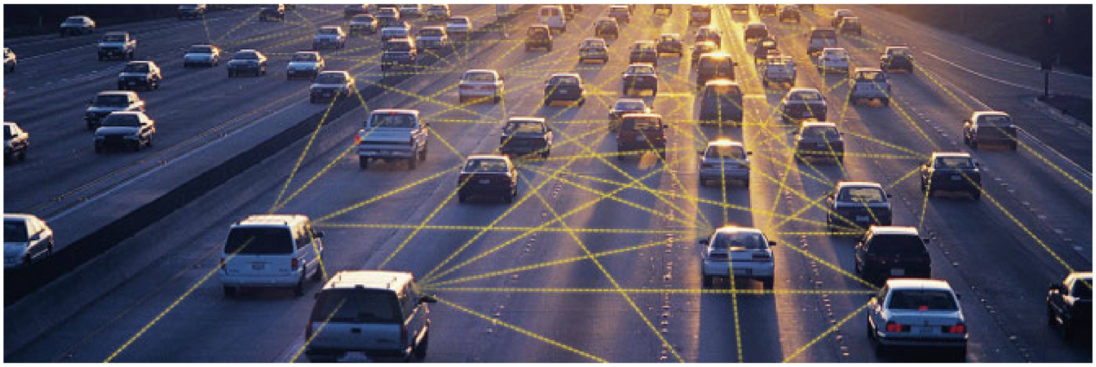


Figure 3.3: Representation of vehicle communication.

- **Driver modelling.** Each vehicle has a different range of accelerations and each user demonstrates behaviours such as aggressiveness with different levels of indicators. Multiple measurements allows adjusting thresholds on the behaviour indicators in order to improve an analysis to adapt to the vast majority of people.
- **Data analytics.** Globalization has promoted the importance of managing data massively in terms of statistics as an information source which is nowadays essential to any institution. The field of data analytics is on constant evolution since the last years. On the specific topic of vehicles, inferring massive statistics on driving behaviours is of interest to insurance companies or governments. The application of "Big data" techniques to data collected by DriveSafe will also be investigated in future work.

Chapter 4

Ahead Vehicle Detection and Tracking

In the last years, researchers and manufacturers have significantly progressed in the development of algorithms and systems that are able to perceive the vehicle environment with inference capabilities that are close to human ones. On the one hand, systems that learn about the static environment (i.e. road markings, traffic signs) have been studied in the state of the art with remarkable results [28]. On the other hand, the dynamic environment (i.e. pedestrians, vehicles) is still a challenging aspect, due to the high variability of the objects to be avoided.

Over the last decades, sensors like RADAR and LiDAR have been widely studied as a solution to this problem. However, the high cost and space constraints of these sensors have situated computer vision as one of the most common alternatives [29]. Cameras have become cheaper, smaller and of higher quality than ever before. In addition, computational costs associated with computer vision have been reduced due to the improvements in processing units.Specifically, today smartphones have the computing capabilities of a full computer from few years ago and a high market penetration. These devices provide good embedded units to solve computer vision problems because of their integrated cameras and their powerful processing capabilities.

With the aim of expanding the analysis about dangerous behaviours (e.g. tailgating), we present an algorithm for ahead vehicle detection and tracking that is integrated in DriveSafe application (see Fig. 4.1). Our proposal is based on a multi-scale approach that takes into consideration the road geometry to overcome the computational constraints. The algorithm is evaluated on a publicly available motorway dataset [30], demonstrating its viability for Advanced Driver Assistance Systems (ADAS) and autonomous driving applications.

This chapter describes the algorithm as a single module independent of DriveSafe App and suitable for any smartphone, as it supposes a contribution in the wide field of computer vision and it has several applications other than behaviour analysis.

4.1. Related work

Vision-based object detection has been widely studied over the last years. One of the key works that supposed a breakthrough is the Viola-Jones algorithm [31], which is based on a sequential classifier with Haar-like features that demonstrated real-time performance on the face detection problem. Since then, researchers have proposed several approaches based on multiple classification algorithms (SVM [32], AdaBoost and variants [33], [30]) and varied features (Haar [34], LBP [30], HOG [32], ICF [33] and ACF [35]).

On the specific research area of vehicles, detection in static images is a widely studied topic. The

Figure 4.1: DriveSafe App running in a real environment.

recent appearance of vehicle-annotated motorway datasets (i.e. LISA [36] and TME Motorway [30]) has allowed to evaluate detection performance on real driving environments. Thus, performance is measured not only as a theoretical amount of true positives in a set of images but as real execution in a dynamic motorway environment, where tracking and filtering techniques are as important as detection.

In this way, there have been several recent works that have focused on developing improved tracking algorithms by making minimal changes to the Viola-Jones detection framework. Particle Filtering (PF) has been a widely used technique. An example based on PF is [37], which proposes a full integration between lane and vehicle tracking modules to complement each other. In [30], an algorithm based on Flock of Trackers and a Learn and re-detect module is implemented following TLD method (Tracking-Lerning-Detection). The work in [38] proposes the use of a Probability Hypothesis Density (PHD) filter to track features detected within the bounding box of the vehicle, with some pruning techniques to counteract the high computational requirements of the filter.

Additionally, the motorway environment allows to apply expert knowledge from the road. Several works focus on applying geometrical constraints, mostly to satisfy real-time requirements by pruning the detection search window. For instance, methods like [32] implement an adaptative coarse-to-fine object search to restrict possible scan-ROI positions. This work also applies a multi-scale feature preprocessing stage to award more resolution to distant ROIs and reduce processed pixels over 50 %, similarly to what is done in the present work. The approach in [33] proposes reducing scales per octave depending on uncertainty from tracked object and giving computation priority to near vehicles. Both works use variants of AdaBoost that early reject regions with low object probability: Boosted Cascade in [32] and Soft-Cascade in [33].

With the aim of satisfying both computational and detection requirements, we combine multi-scale approaches with low-computational tracking methods. Simple considerations of the road geometry are taken into account by means of a lane detector. The result is an efficient algorithm that is suitable for smartphones. At the end of this chapter, it is evaluated on a motorway dataset to prove its performance on a real environment.

4.2. System design

This work is focused on developing a vehicle detection and tracking algorithm suitable for smartphones. Hence, a key aspect in the system is the computational efficiency. As seen in the state of the art, scanning a full image with a classifier is an expensive operation. Thus, applying knowledge from the road geometry to prune the search window is more efficient than scanning the image by brute force. Considering any road environment, vehicles in the scene may either appear from behind as near vehicles

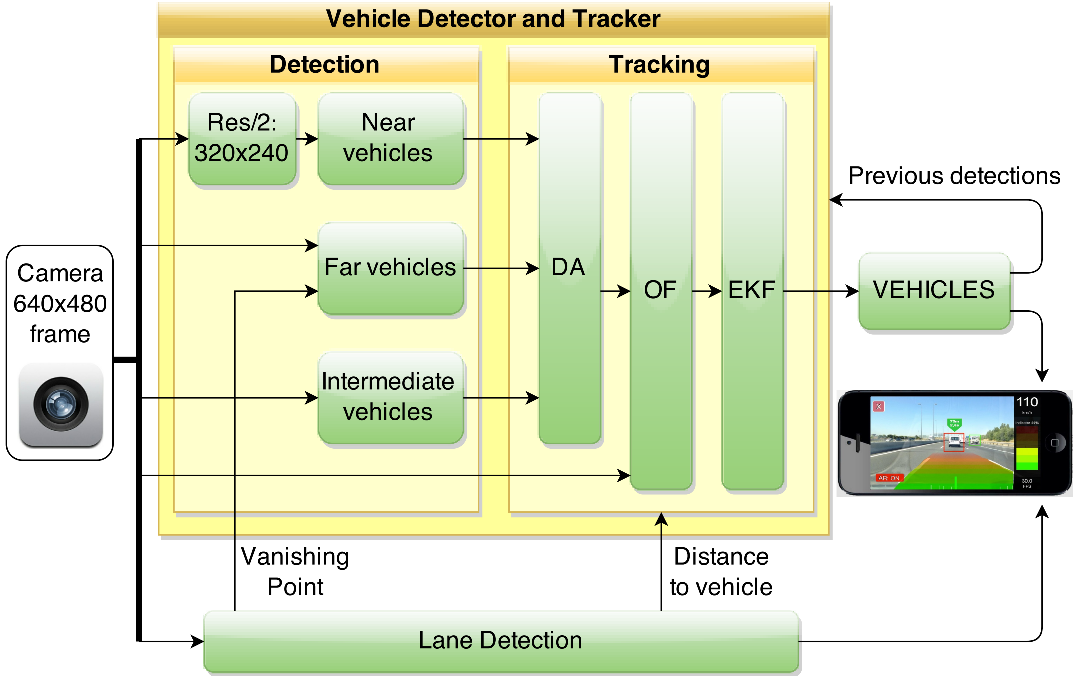


Figure 4.2: Full system diagram of the vehicle detection and tracking module. DA, OF, and EKF stand for Data Association, Optical Flow, and Extended Kalman Filter respectively.

(if they move faster than us) or either be approached from far (if they move slower than us or they are in the opposite direction lane). With this assumption, we propose a multi-scale approach that optimizes the discovery of new vehicles by making use of different detection windows.

The detection can be divided into three main stages, as can be seen in Fig. 4.2. They are based on an AdaBoost classifier. Firstly, a large image patch with low resolution is scanned by the detector to mainly discover near vehicles (Sec. 4.2.1). Secondly, a small image patch in the surroundings of the vanishing point is scanned at a higher resolution in order to discover far vehicles (Sec. 4.2.2). Thirdly, the detector scans specific image patches corresponding to previously discovered vehicles in order to reaffirm detections and to cover those that do not fit in the near nor the far scan windows (Sec. 4.2.3). Fig. 4.3 shows a video-frame where each of the stages is relevant to perform the detection of a specific vehicle in the scene.

Data association between detections in each frame and their corresponding tracked candidates is decided by a simple overlapping formula (Sec. 4.2.4). Position estimation is enhanced with Optical Flow (Sec. 4.2.5). Tracking is performed by an Extended Kalman Filter (Sec. 4.2.6). Lane detection (Sec. 4.2.7), which is performed by DriveSafe App, is seized to enhance the detection by providing estimated distance to the car and road vanishing point.

The whole implementation is oriented towards simplicity in order to fulfil real-time constraints. Thus, the input image is reduced to 640x480, from which all detection windows are extracted. All stages are performed in every frame.

4.2.1. Detector 1: Near window

In this stage, the image resolution is resized to a half resolution (320x240). This extremely reduces computational cost in the AdaBoost detector, although it has two negative consequences. Firstly, the detector recall rate is reduced for vehicles that are farther than a distance (≈30m), as the lower resolution

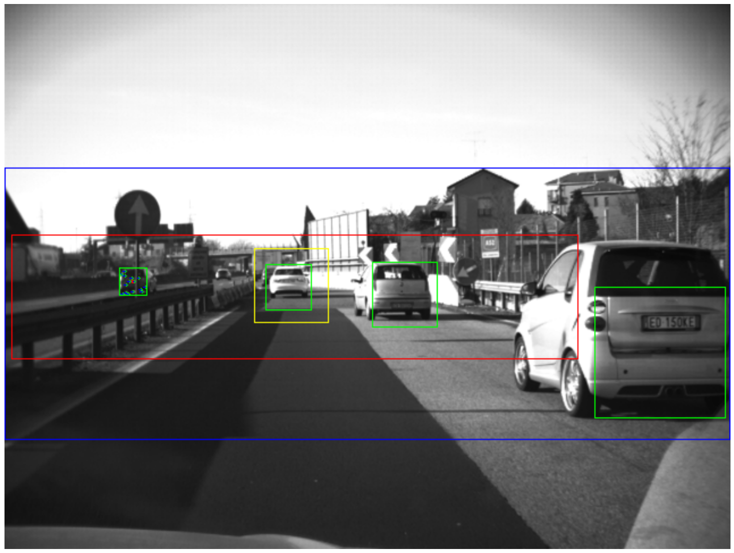


Figure 4.3: Example of the three vehicle detection stages working. Green boxes represent detections, blue the near-window, red the far-window, yellow the intermediate window (vehicle-specific) and the points depicted in blue in the vehicle of the left represent Optical Flow.

diffuses the vehicle features. Secondly, the training process is done with a fixed model size (20 pixels in our case), and then objects with a pixel width lower than this threshold will not be detected by the classifier. Therefore, the function of this detection ROI is only to detect near vehicles, leaving those that are further than 30 meters to the intermediate or far window.

Additionally, the sky is filtered out to avoid unnecessary computation. Considering that the camera will be perpendicular to the ground, the horizon provided by DriveSafe is approximately situated at half of the image. With the aim of leaving a margin for near cars or trucks, only about 30 % of the top is removed. About 20 % of the bottom of the image is also removed in order to avoid the car bonnet (estimated by DriveSafe) and a small part of the nearby road that does not affect detection. Thus, a total of around 50 % of the vertical size of the image is removed, reducing considerably the computation cost of this stage.

4.2.2. Detector 2: Far window (zoom)

The far window covers the deficits of the previous stage for detecting distant vehicles. Thus, a small patch is extracted from the input image in the neighbourhood of the vanishing point, which is provided in our system by the lane detector module of DriveSafe (Sec. 4.2.7). This approach using the vanishing point of the image supposes an improvement over using a fixed window in the half of the image in cases of curved roads.

The size of this window must be small enough to have a light computational cost and large enough to fit most far vehicles Therefore, this size is experimentally set to 500x110 pixels (see red rectangle in Fig. 4.3).

Considering that this far window is overlapped within the near one, this region will be scanned twice by the detection. Thus, some vehicles could be detected by both detectors. In these cases, the score of the AdaBoost detector (a weighted sum of the weak classifiers) is used to determine which of both detections is predominant.

4.2.3. Detector 3: Intermediate window

Once a vehicle has been discovered by any of the previous windows, a third detection is performed on the surroundings of the last position of the car. This ensures re-detection independently of road position, covering intermediate areas where the other windows have deficiencies. Thus, it reaches all those vehicles that have moved far enough to not be correctly detected by the near window but are not far enough to fit inside the far window.

The intermediate window is vehicle-specific. It is focused on the surroundings of the previously known vehicle position. An extra size is given by a margin of half the width and height on each of the sides of the previously detected bounding box. The image used to extract this window is the full 640x480 one. As it will be seen in Sec. 4.3.3, the computing cost of this window is extremely low.

4.2.4. Data association

In order to associate new detections in the current frame and tracked candidates, an overlap matrix is computed as follows:

$$overlap = \frac{area(BB_i \cap BB_j)}{area(BB_i \cup BB_j)} \tag{4.1}$$

where BB is the bounding box of new vehicle i and tracked candidate j. Vehicles are associated if the overlap is higher than a certain threshold.

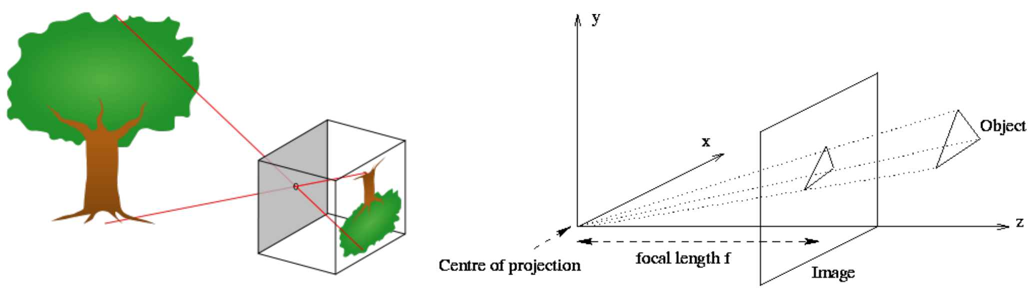
When a new vehicle is discovered and it does not overlap with any previous candidate, it is saved as a new candidate if it is detected by either the near-window or the far-window at least twice in the last three frames. This significantly reduces false positives and avoids that these detections are kept in the system by the tracking algorithm. When a candidate does not have a new detection associated in the current frame, this is moved using optical flow of local features.

4.2.5. Optical Flow

Local features from detected vehicles are used to enhance tracking. On detection, a small image patch containing the vehicle is saved. This image patch is used to compute features that are searched in the next frame in the proximity of the last known position. Thus, corners are computed by Shi-Tomasi method and are tracked using the optical flow obtained by Lucas-Kanade algorithm [39]. The median of the flow from all found points is used to compute the vehicle motion, which is used to correct the previously known position of an unpaired candidate in the current frame. When less than ten points have been tracked by the algorithm, the candidate is considered not valid and removed. This step has low computational cost, as the corner search happens only in a reduced region.

4.2.6. Extended Kalman Filter

Tracking is performed with an Extended Kalman Filter (EKF). While the measurements obtained by the camera are in pixels (monocular camera), the objects that are tracked are in 3D coordinates. The pinhole camera model (see Fig. 4.4) is used to set the following 3D states of each candidate: lateral position (x), longitudinal position or object distance to camera (d), vehicle width (d), and their respective derivatives. Vehicle height is not used because the detected vehicle pattern is square. 3D distance measurement is calculated by the lane detection module of DriveSafe.



(a) (b)

Figure 4.4: Pinhole Camera Model. a) Pinhole camera representation. b) Coordinate axes in the model.

The equation that represents the pinhole camera model is:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{z_0} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{4.2}$$

where u and v are positions in the 2D image in pixels, f is the focal distance of the camera, and x_0, y_0 and z_0 are positions in the 3D coordinate axes.

The states and measurements vectors are considered as follows. All variables of the states, enumerated in the previous paragraph, are in meters. About the observations or measures, u is the lateral position in the image in pixels, d is the distance estimated by the lane detector in meters (to simplify calculations) and w_p is the vehicle width in pixels.

$$States(X) = \begin{bmatrix} x & d & w & x' & d' & w' \end{bmatrix} \tag{4.3}$$

$$Measures(Z) = \begin{bmatrix} u & d & w_p \end{bmatrix} \tag{4.4}$$

Therefore, the pinhole model is used for Observation (measurement matrix H transforms observations into states to correct the prediction). The Jacobian of the measurement matrix H is calculated due to the non-linearities of the model, where all the variables depend on the distance to the object. k is a constant that considers the camera parameters (i.e. size of the CCD sensor) to correctly transform pixels to meters:

$$J_H = \begin{bmatrix} {}^{f \cdot k}\!/_{d} & {}^{-f \cdot k \cdot x}\!/_{d} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & {}^{-f \cdot k \cdot w}\!/_{d} & {}^{f \cdot k}\!/_{d} & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

And the constant velocity model is used for Transition. Transition matrix F updates (predicts) states from one instant of time to another:

$$J_F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

This filter is essential to keep a stable detection and avoid the flicker produced when the detection is noisy or when there is a change between windows in the multi-scale approach.

4.2.7. Lane detection

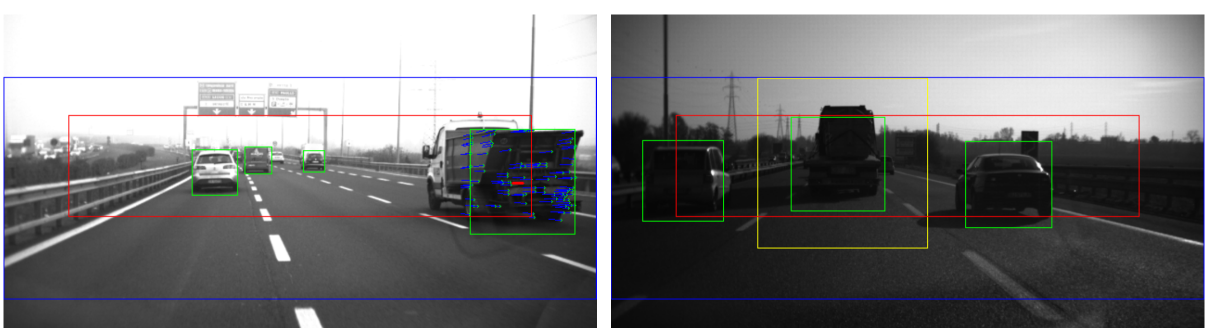
Lane detection is carried out by DriveSafe App to enhance vehicle detection by providing road information. Although it is not strictly needed for vehicle detection, it can enhance its functioning by providing road information. In our algorithm, it specifically provides the vanishing point for positioning the far window and the estimation of distance to ahead vehicles for tracking and filtering. In short, the lane detection system is composed by a Canny algorithm [40] to detect edges, Hough Line Transform [41] to extract lines and a Kalman Filter to stabilize the measurements. Reader may refer to [7] in order to learn more about the implementation details.

4.3. Results

A Gentle AdaBoost classifier is trained based on LBP. Haar was also tested and demonstrated much slower performance with no significant improvement. HOG, ICF and ACF were discarded due to computational constraints. The classification is performed on 20x20 pixel patterns. This defines the model size, and thus the minimum vehicle width that can be detected in the image. This value affects at the image resolution used by the detector (640x480). In the charts, pixel values are translated to 1024x768 for comparison with the original dataset, so the minimum vehicle width is shown as 32 pixels.

All images used for the training are publicly available and they correspond to a mixture between public datasets. Positives are only obtained from GTI dataset [42], corresponding to 3425 images of rears of cars and trucks. Negatives are obtained from multiple datasets: GTI [42], KITTI [43], Caltech Cars (Rear) background [44] and GRAZ-02 [45] (only images of background without cars), forming a total of 13868 negatives.

To the best of our knowledge, there are only two publicly available vehicle-annotated datasets for motorway video sequences: LISA Vehicle Detection [36] and TME Motorway [30]. The first includes 2 motorway sequences for a total duration of one minute, while the latter is formed by 28 motorway clips for a total of approximately 27 minutes (30000+ frames). We present the results on the larger one (TME), as it allows evaluating the performance in a real motorway situation. Recent works on vehicle detection have presented results on it: [30], [32], [33] and [38] (this one only presents results about tracking error). It contains two subsets: “Daylight”, which is larger and its light conditions might be more common in daily situations, and “Sunset”, which was recorded in challenging conditions where the sun is low and its light significantly affects visibility, allowing the evaluation of the algorithm robustness. Fig. 4.5 shows the image differences between both subsets.



(a) Detection on “Daylight” subset

(b) Detection on “Sunset” subset

Figure 4.5: Representative images of detection on each of the dataset’s subsets. a) shows a clear image on an average day, while b) shows the challenging shadowing effects produced during sunset.

4.3.1. Detection performance

In Figs. 4.6 and 4.7 we present the evaluation statistics collected for the algorithm performed on the dataset. Precision is computed as $\frac{TP}{TP+FP}$ and recall rate as $\frac{TP}{TP+FN}$, where TP, FN and FP are respectively the number of true positives, of false negatives and of false positives, matched between system output and the approximate ground truth (GT'). As explained in [30], the dataset's approximate ground truth (GT') was obtained from laser scans, hence the reliability limitation beyond 60-70 meters, when less than 3 laser reflections per vehicle become available. The results are presented in the same format as the original paper for comparison, with the same evaluation process.

Results show that precision remains over 90 % for both subsets (Daylight and Sunset). Recall rate is over 95 % for near cars and it remains very high until 60m. The difference in performance between trucks and cars corresponds to the lower proportion of truck samples that is used in the training set. Ideally, cars and trucks should be detected by different classifiers due to the remarkable model difference, but this would not be a computationally efficient solution. Thus, both are used together to train the same classifier, which results in an acceptable overall performance.

Despite the fact that the GT' is less reliable over 60m, the algorithm detection rate decays over this distance due to the low resolution that is kept for computational constraints. Our work is focused on providing high detection rates within a reasonable distance. Ahead vehicles under 60m have the most impact on driving behaviour, and are the ones to be utterly considered by ADAS or autonomous driving algorithms.

As can be seen in Fig. 4.7, detection performance remains similar in the more challenging "Sunset" subset. This demonstrates that the algorithm is robust against light variations. Additionally, further tests in a real driving environment have also demonstrated robustness under severe rain conditions, although we cannot provide quantitative results due to the lack of ground truth.

4.3.2. Comparison with related works

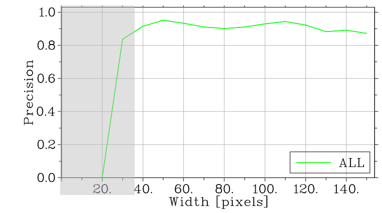
The objective of this work was not to improve the detection performance of an existing system, but obtain similar results with state-of-the-art works, considering the smartphone computational constraints. In Fig. 4.8, we show our results compared to state-of-the-art methods that made use of this dataset: [32] (Boosted Cascade + Haar + LRF + Road constraints), [30] (WaldBoost + LBP + FoT) and [33] (SoftBoost + ICF).

As can be seen in Fig. 4.8a and 4.8c, the precision of our algorithm is slightly lower than other results because high recall has been preferred in the trade-off between precision and recall, as we give more importance to detect all cars in a range and noisy detections are easier to filter with further lane analysis. Fig. 4.8b and 4.8d show that recall rate is higher than other state-of-the-art works until 60m. Additionally, the results for "Sunset" show that our algorithm is the least affected by bad light conditions.

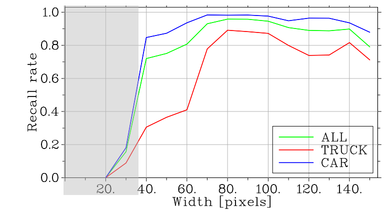
4.3.3. Computing performance

Table 4.1 shows the computation time for each of the modules of the detection algorithm for three different devices (iPhone5, iPhone6, and iPhone Simulator). The most expensive stage is the far-window due to its high resolution, even though it covers a smaller region. This demonstrates the advantage of multi-scaling, as a single full-resolution window would be inviable.

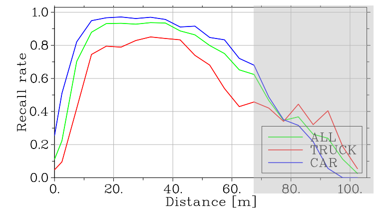
The detector runs inside our DriveSafe application, simultaneously with the rest of its ADAS modules. The hermetic character of iOS does not allow to isolate the application from other unrelated background processes, so it was not possible to guarantee full CPU dedication to the algorithm. This produces



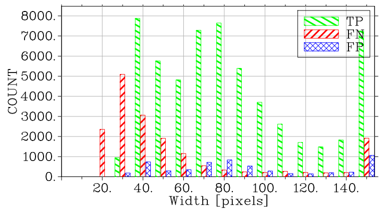
(a) Precision in function of width



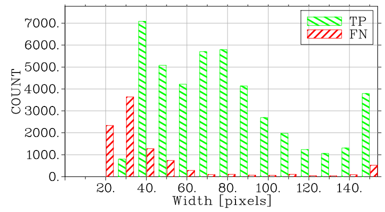
(b) Recall rate in function of width



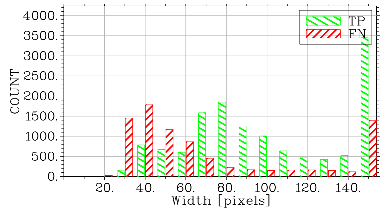
(c) Recall rate in function of distance



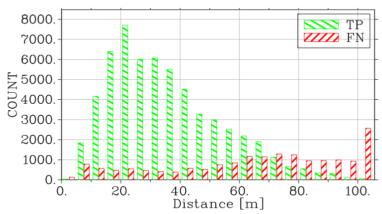
(d) All detections, grouped by width



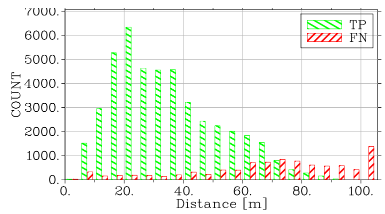
(e) Car detections, grouped by width



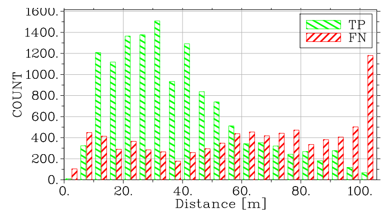
(f) Truck detections, grouped by width



(g) All detections, grouped by distance

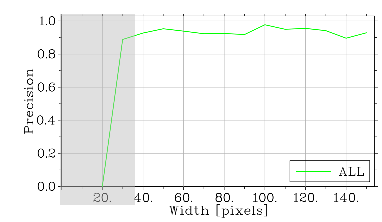


(h) Car detections, grouped by distance

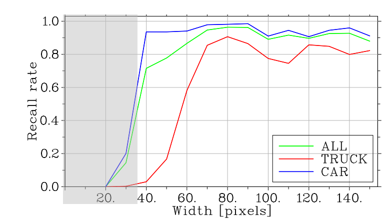


(i) Truck detections, grouped by distance

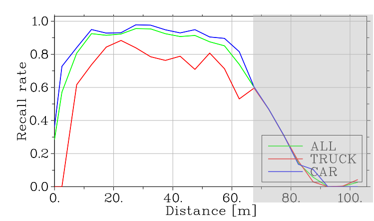
Figure 4.6: Detection algorithm evaluated on the “Daylight” subset of the TME Dataset. A grey box is placed over the chart region where the GT’ is not considered reliable due to a laser limitation. Vehicle pixel width corresponds to a resolution of 1024x768, for comparison with original charts.



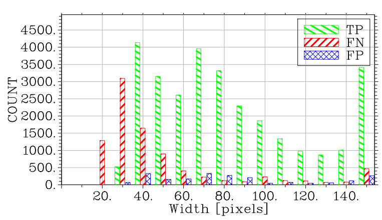
(a) Precision in function of width



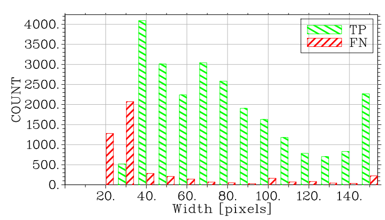
(b) Recall rate in function of width



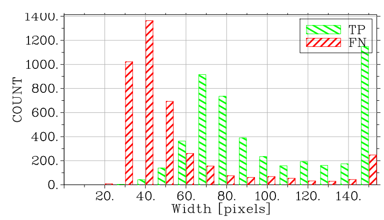
(c) Recall rate in function of distance



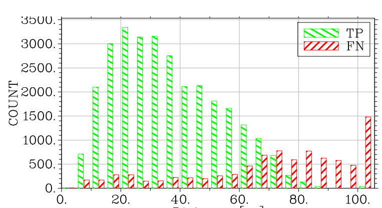
(d) All detections, grouped by width



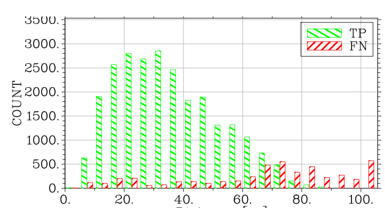
(e) Car detections, grouped by width



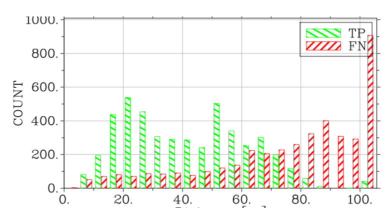
(f) Truck detections, grouped by width



(g) All detections, grouped by distance

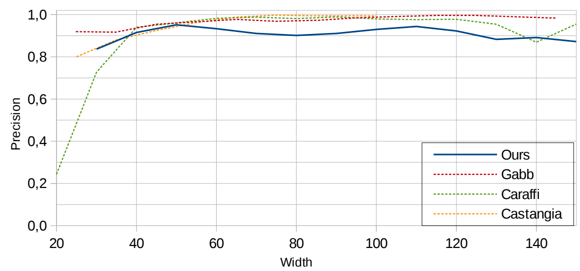


(h) Car detections, grouped by distance

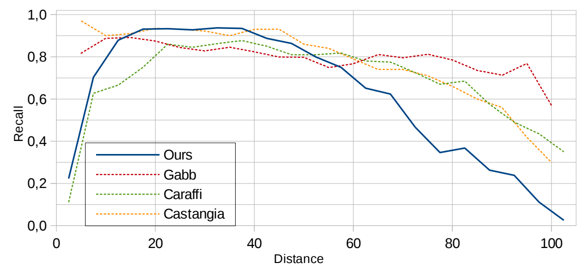


(i) Truck detections, grouped by distance

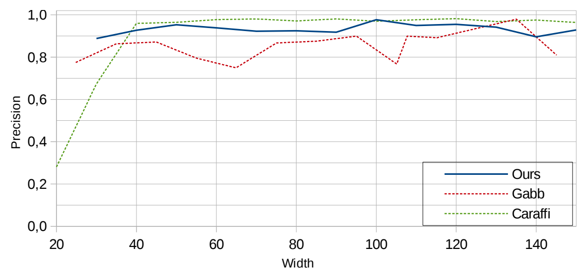
Figure 4.7: Detection algorithm evaluated on the “Sunset” subset of the TME Motorway Dataset.



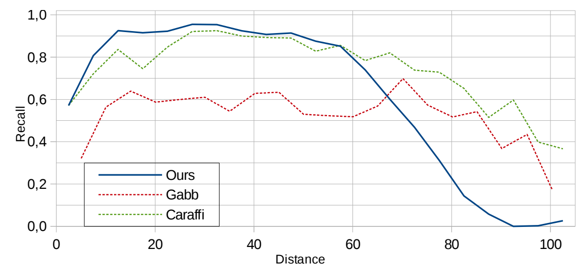
(a) Precision in function of width (Daylight subset)



(b) Recall rate in function of distance (Daylight subset)



(c) Precision in function of width (Sunset subset)



(d) Recall rate in function of distance (Sunset subset)

Figure 4.8: Comparative charts of state of the art results for “ALL” (cars and trucks) between our approach (“Ours”), results presented in [32] (“Gabb”), in [30] (“Caraffi”) and in [33] (“Castangia”). Results of Sunset subset are not available in [33].

a slight decay in the expected performance. Nevertheless, it supposes a more realistic evaluation, since the results obtained correspond to an application running in the smartphone OS along with other several routines (e.g. call management), which is the case for the average user.

These results represent the total time needed by one CPU core to process one frame. Considering the camera has a rate of 30 fps, not all frames are processed. In DriveSafe App, the detection algorithm runs as a medium-priority thread providing detections. From experimental tests in a motorway environment, we have inferred that 5-10 fps is enough for a robust detection performance. Running the algorithm on the TME dataset with lower frame rates (5 and 10 Hz) by skipping frames has also produced similar results to those presented in Fig. 4.6 and 4.7. The worst case scenario are vehicles that overtake or brake suddenly near the camera sides, thus producing the largest motion in image pixels. Even in these cases, the relative speed between the detected vehicle and the camera carrier has to be immensely high to produce a loss in the tracking process.

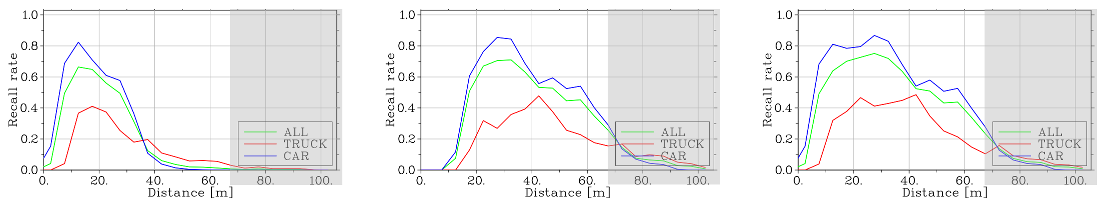
Results are also provided for an iPhone 6 simulator run by an Intel Core i7@2.2GHz processor for comparison with other works. Although the simulator supposes a significant computational overhead and the algorithm would achieve higher frame rates as a standalone code, it doubles the standardized real-time rate of 30 fps. This demonstrates its portability to other devices such as in-vehicle computers.

4.3.4. Detection windows comparison

Fig. 4.9 depicts the performance of each of the two discovery windows (near and far) separately. The first detector covers from a small distance until approximately 30m (Fig. 4.9a). The second one covers from 20m until 60m (Fig. 4.9b). The algorithm with both detectors operative shows to cover both ranges successfully (Fig. 4.9c). These results only depict near and far window, lacking intermediate and tracking stages. The difference in performance between these results and those presented in Fig. 4.6 is covered by the rest of the stages.

AVERAGE COMPUTATION TIME [ms]			
Module	iPhone 5	iPhone 6	Simulator (i7)
Near-window	48	26	6
Far-window	70	41	8
Intermediate-win.	6	4	1
Optical Flow	3	2	<0.5
Extra operations	5	3	<0.5
TOTAL	132	76	16
FPS	7.6	13.2	62.5

Table 4.1: Average computation time for detection modules.



(a) Recall rate by distance for near-window

(b) Recall rate by distance for far-window

(c) Recall rate by distance for algorithm with both near and far windows operative

Figure 4.9: Comparison of detections covered by windows near, far and both together on the subset “Daylight”. Neither of these results have any kind of tracking.

4.4. Conclusion

We have presented an ahead vehicle detector and tracker that can run in real time on an iPhone, in parallel with other ADAS processes such as a lane detector. A multi-scale approach and geometrical considerations have been used to overcome the computational constraint. The result is a versatile algorithm that can be implemented on any smartphone and performs well on motorway environments without the requirement of large training sets. Experiments on a publicly available motorway dataset demonstrate that our detection performance is similar to state-of-the-art works and it is robust against light changes. Its low computational cost allows an efficient integration in ADAS or autonomous vehicles.

Future work will involve producing a higher-level of driving behaviour analysis based on the vehicle detection and the techniques applied in [7]. Improvements of the detection and tracking algorithm could be based on a higher integration of the lane and vehicle modules to complement each other in a more complex way. In addition, other preprocessing stages could be added for minimizing issues associated with shadowing or sunset conditions in vehicle detection, such as the illumination invariance techniques presented in [46].

Chapter 5

Driver Analysis

The main aims of DriveSafe App are to evaluate driving behaviours and to give the corresponding feedback to drivers to encourage good driving practices. In this master thesis, the set of variables that DriveSafe can sense has been augmented to enhance the driver analysis. Aside from the new sensing capabilities such as the ahead vehicle detection, this work has also introduced the connection to a road information online API (OpenStreetMap) to produce a fusion between sensors and map information. This way, the sensed data is also used to locate in a virtual map (via GPS coordinates) and thus obtain further information from that location (e.g. road maximum speed) to enrich the global analysis.

This chapter describes how all the information gathered by the smartphone is transformed into driver analysis and seized in the best possible way. Firstly, Sec. 5.1 describes how the measurements are arranged in a variety of indicators that serve as specific units to characterize behaviours. Secondly, these indicators are gathered to produce a set of numeric scores (Sec. 5.2), with two purposes: simplify things numerically and give feedback to the user. Thirdly, they are also gathered to profile the driver by estimating a behaviour model Sec. 5.3. Thus, the whole architecture of DriveSafe about driver analysis has been updated and rearranged to cover all the new variables obtained by DriveSafe.

Sec. 5.4 describes some changes in the interface that have been introduced to give more driving feedback to the user. Finally, Sec. 5.5 presents a complete evaluation of the presented indicators and behaviour profiles on a set of varied drivers and tests.

5.1. Indicators

A smartphone is able to sense plenty of information as it counts with a rear camera, a microphone, inertial sensors, GPS and communication capabilities. However, this information is captured by the phone raw, unprocessed. As a prior stage to evaluating the driver, this information must be arranged in some sort of indicators that describe the driving process as a group of events or numeric facts. This way, the more different indicators that we are able to extract from the sensors on the device, the richer will be the analysis that can be produced. This section describes all the driving indicators that are wrapped from every capability that DriveSafe is able to seize from the smartphone.

5.1.1. Inertial sensors

One of the richest type of measurements to evaluate driving are inertial sensors. Nowadays, almost every smartphone carries a set of accelerometers in every axis (X,Y,Z) and gyroscopes (pitch, roll, yaw). In the case of the gyroscopes, they are only used in the calibration process to position the device per-

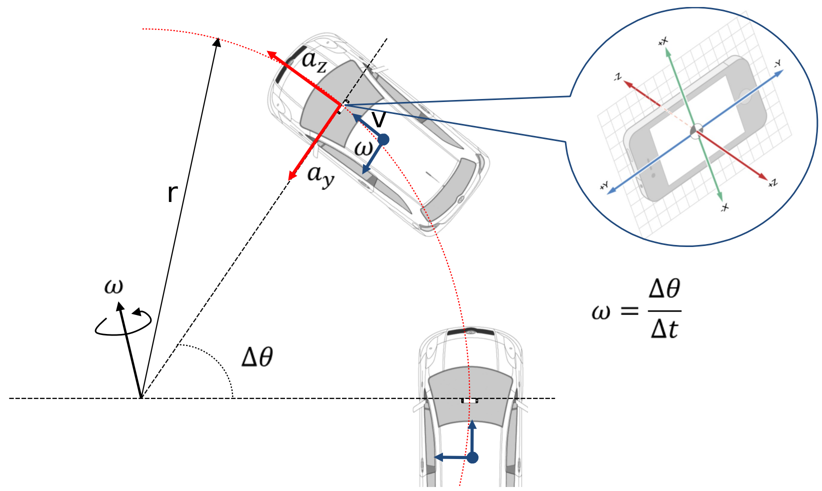


Figure 5.1: Coordinate frame of iPhone vs vehicle.

pendicular to ground, aligned with longitudinal and lateral vehicle axis. Accelerometers, however, are an excellent way of evaluating the brusqueness of movement during driving. Sudden accelerations, brakings or turnings often imply a distraction, or an aggressive behaviour if they are regular.

As it was seen in Sec. 2.3 and is depicted in Fig. 5.1, the phone is positioned in a way such that Z axis is aligned with the vehicle's longitudinal axis, and Y is aligned with the vehicle's lateral axis. Thus, a sudden positive increase in Z accelerometer represents an acceleration, where abrupt peaks may indicate aggressive increases of velocity. A decrease in the same accelerometer represents a sudden deceleration, which may be an indicative of harsh braking, and therefore indirectly of not retaining a minimum distance to the vehicle ahead. Finally, high increases or decreases in Y axis are indicatives of excessive velocity in left or right turns, which may result in the vehicle loosing traction.

As it was explained in the work that presented DriveSafe [7], the thresholds set to detect these events were extracted from [5] and expanded to three different levels for each one in order to get a better evaluation of the driver's behaviour. These levels are represented in Table 5.1.

Event Type	Threshold sensitivity		
	Low	Medium	High
Acceleration	$0.1g < a_z < 0.2g$	$0.2g < a_z < 0.4g$	$a_z > 0.4g$
Braking	$-0.1g > a_z > -0.2g$	$-0.2g > a_z > -0.4g$	$a_z < -0.4g$
Turning	$0.1g < \|a'_y\| < 0.2g$	$0.2g < \|a'_y\| < 0.4g$	$\|a'_y\| > 0.4g$

Table 5.1: Distraction event thresholds.

Pre-proccesing. Accelerometer data is sampled at a rate of 100 Hz. The raw data from the iPhone contains significant amounts of noise from the vibrations onboard the vehicle. Thus, this signal is cleaned using a Kalman filter with a state vector formed by the three components of the accelerometer (a_x, a_y, a_z). The filtered features prove to be highly correlated with the vehicle movements.

Event detector. We use a triple threshold that comprises a minimum absolute acceleration value, a minimum time period during which this value is exceeded and a minimum longitudinal velocity of 50 Km/h. Moreover, each event is identified with its intensity (low, medium, high) depending on the thresholds in Table 5.1. When an event is activated, a hysteresis period is enabled to account for potential activations in the near future.

As it is remarked in [5], none of the currently available applications for the assessment of driving behavior consider the dependence of the event counts with the road geometry. To solve this problem we propose to decouple the lateral acceleration due to the road curvature from the one caused by wrong driver movements. When a vehicle makes a turn, it experiments a centripetal force, which has its direction orthogonal to the direction of movement of the vehicle and toward the center of the turn. This centripetal force generates a centripetal acceleration, a_y^c, also pointing towards the center of the curve. Assuming a turn following a perfect circle, the centripetal acceleration (a_y^c) can be obtained by using the angular speed (ω), the tangential velocity (v) and the radius of the turn [47].

$$a_y^c = \frac{v^2}{r} = r \cdot \omega^2 = \omega \cdot v \tag{5.1}$$

Based on Fig. 5.1, it can be seen that vehicle and iPhone have the same radii, then, they will have the same centripetal acceleration. Taking into account that (v, ω) can be estimated each second from the GPS, we have a coarse estimation of the centripetal acceleration of the vehicle due to the road. Subtracting the lateral acceleration measured by the iPhone from the centripetal acceleration of the vehicle, we estimate the lateral vehicle acceleration due to defective driving maneuvers ($a_y'= a_y - a_y^c$). Inspired by this simple yet useful physics observation, we have reached to solve an important gap of the state of the art.

5.1.2. GPS

The main indicators that can be obtained from the GPS sensor are:

- **Speed:** The vehicle speed in kilometres/hour is an essential measurement to evaluate overspeed or to relate distance to ahead vehicle with the estimated time of impact.
- **Course:** The GPS course allows detecting curves to decouple sudden turnings (as explained in previous section).

Although GPS values are measured instantaneously every second (1Hz), in some cases it is necessary to predict future GPS position, for instance to request information from APIs such as OpenStreetMap (see Sec. 5.1.5). These APIs delay the response on the range of various seconds (between 2-5 seconds in our tests), so it is necessary to request information on a predicted position to solve this delay. Thus, the future GPS position is computed as follows.

In an (East, North) local frame, the position after the time interval is :

$$x = speed * sin(heading * \pi/180) * dtime/3600; \tag{5.2}$$

$$y = speed * cos(heading * \pi/180) * dtime/3600; \tag{5.3}$$

where $speed$ is the vehicle velocity in km/hour, $heading$ is the GPS course (i.e. backwards angle from azimuth 0º in degrees), π is the Pi constant (3,14159), and $dtime$ is the time interval from the start position in seconds.

From there, we compute the new position in the WGS84 frame (i.e. latitude and longitude):

$$lat = lat0 + 180/pi * y/Rt; \tag{5.4}$$

$$lon = lon0 + 180/pi/sin(lat0 * pi/180) * x/Rt; \tag{5.5}$$

where $lat0$ and $lon0$ are the initial coordinates in degrees, and Rt is the Earth radius in kilometres (6378.137 km).

5.1.3. Lane detection

Lane detection allows the extraction of two main indicators:

- **Lane drifting:** Driver's tendency of moving inside current road lane. This may indicate inattention due to the incapacity to keep in the centre of the lane. Further described in Sec. 5.1.3.1.
- **Lane weaving:** It measures driver's tendency to switch between lanes. It is evaluated as the amount of lane departures and their wilfulness. A high number of involuntary lane departures might be an indicative of drowsiness. Further described in Sec. 5.1.3.2.

5.1.3.1. Lane drifting

Lane drifting estimation, as was explained in Sec. 2.4.4, is based on the indicator called Lanex (fraction of Lane exits), which is a measure of driver's tendency to exit the lane [26]. This indicator has remained unmodified from the one developed in the original version of DriveSafe [7]. It is defined as the fraction of a given time interval spent outside a virtual driving lane around the center of 1.2 m width, calculated by applying windowing techniques over the lateral position of the vehicle (x_0) during 60 s. Fig. 5.2 show an example of lane drifting. This information is depicted in DriveSafe by road lane submarkers at the bottom of the image together with a coloured position indicator.

(a) (b)

Figure 5.2: Example of lane drifting. (a) Driver keeps inside the right lane centre margins. Position indicator remains green. (b) Driver position is biased towards the right. The position indicator becomes orange.

5.1.3.2. Lane weaving

Lane weaving evaluates the amount of lane changes that are produced during driving. In DriveSafe, we relate this measurement with the wilfulness of these changes, as involuntary lane departures may indicate inattention or drowsiness.

Wilfulness of lane departures was firstly evaluated in DriveSafe via the inbuilt microphone by analysing patterns in the sound to infer whether the driver had pressed the turn signal or not. This demonstrated effective for a specific vehicle, but it had problems with the significant differences of the turn signal between vehicle manufacturers. Additionally, opening the vehicle's windows, the high noise of some vehicle's motors, or keeping a high volume of the radio affected significantly the detection. On a second approach, an OBD connector (i.e. vehicle port that provides some on-board diagnostics) was tested as a way to infer whether the driver had pressed the turn signal on a lane departure. Once again, it is very effective on those vehicles that output this information via the OBD connection. However, this port is not very accessible: the connection standards for each vehicle model are very different and are not provided by the manufacturer, forcing reverse engineering techniques. These differences also produce that information such as the turn signal is not available in several models, as it depends on limitations set by the manufacturers. Additionally, the driver would have to buy an OBD-II connector, supposing an extra cost.

(a) (b)

Figure 5.3: Process of a changing a lane. (a) Intant previous to a lane departure. Blue indicator depicts position close to the lane side. (b) Instant posterior to a lane departure, detected as voluntary.

Finally, wilfulness in lane departures has been solved in the following way. Normal and voluntary lane departures are performed within a short time window. On the other hand, involuntary lane departures that are due to inattention are normally performed in a long time period, as vehicle exits lane slowly without driver's notice. Following this reasoning, we have made use of the road lane separation in submarkers to detect an imminent lane departure and start a timer that counts the time spent between its start and finish. As Fig.5.3a shows, position indicator becomes blue when the position is close to a lane change. We have measured in several tests the time that is taken in that blue region prior to a lane change, and it is normally around 1 second, with a deviation of between $0{,}30$ and $0{,}50$ seconds. However, each driver performs lane changes in a different way and with different speeds. Therefore, to solve these differences and avoid setting a strict threshold, lane weaving is defined by DriveSafe in the following way:

1. The first 10 lane departures are normally produced close to the beginning of the route, so they are assumed to be voluntary and performed in a normal state.

2. The time within the lane side (blue region) is measured on those 10 departures and the application estimates their mean and standard deviation.

3. Henceforth, it can be considered that any lane departure performed within $mean \pm std\_dev$ is done inside normality. Thus, the final time threshold to infer whether a lane change is voluntary or involuntary is set as:

$$TH_{LD} = mean + 2 \cdot std\_dev \tag{5.6}$$

4. After this point, each lane departure is measured by the time spent on the lane side and it is evaluated against the threshold. If it is performed in less time than TH_{LD}, it is considered voluntary. Otherwise, it is considered involuntary and indicates inattention.

5.1.4. Ahead vehicle detection

In DriveSafe, we have made a great emphasis in developing a vehicle detection and tracking module, as the one presented in Chapter 4, due to the high importance that surrounding vehicles have on driving. As the surrounding dynamical environment, other vehicles affect driving, as they constrain the range of movements that one can perform on the road. For instance, a sudden braking may be produced when driver sees that ahead vehicle's is suddenly braking, or one cannot turn to left lane if there is already a vehicle on that lane. Furthermore, the way of behaving around other vehicles is different for each driver and it may give clues about behaviours such as aggressiveness. Actions such as tailgating (i.e. following ahead vehicle too closely) is a dangerous conduct that is even considered illegal and punishable in some jurisdictions (e.g. in the German one there are fines up to 400 € [48]).

Therefore, vehicle detection is an essential sense in a platform dedicated to driver analysis, whether it is to infer a behaviour model or to encourage good driving practices by alerting and scoring the driver. Coupling this sense with the lane detection allows to obtain several indicators that are useful in driver analysis:

- **Distance to ahead vehicle:** Lane estimation allows to estimate whether an ahead vehicle is in the same lane as the DriveSafe user. In those cases, the distance that the user keeps to ahead vehicle gives rich information about his behaviour. It is possible to estimate the distance to ahead vehicle even though the image is captured by a monocular camera, by applying the pinhole camera model and considering an approximate lane width.
- **Time of impact to ahead vehicle:** Distance coupled with GPS speed allows to better weigh its relevance. For instance, keeping 50m to ahead vehicle is much more relevant in terms of aggressiveness if the speed is 120Km/h than if it is 50Km/h. Thus, in some cases it is more interesting to treat directly the time of impact, in seconds, than the raw distance.
- **Amount of vehicles detected:** Although the camera can only reach ahead vehicles and until a certain distance, keeping a record of the vehicles detected in a time window provides an estimation of traffic density. A driver gets clearly influenced by the amount of surrounding vehicles and he does not behave equally when there is high traffic than when he is alone in the road.
- **Overtakings:** A high frequency of overtakings (i.e. passing other vehicles on the road) may imply impatience and a certain degree of aggressiveness.

(a) (b)

Figure 5.4: Examples of vehicle detection. (a) Driver keeps a relatively close distance (19.5 meters, 0.6 seconds) that is depicted in yellow colour. The module also detects 6 vehicles in the same frame, indicative of high traffic density. (b) Driver is at a very close distance to ahead vehicle, depicted in red colour. DriveSafe also emits an alerting sound on these dangerous cases.

5.1.5. OpenStreetMap

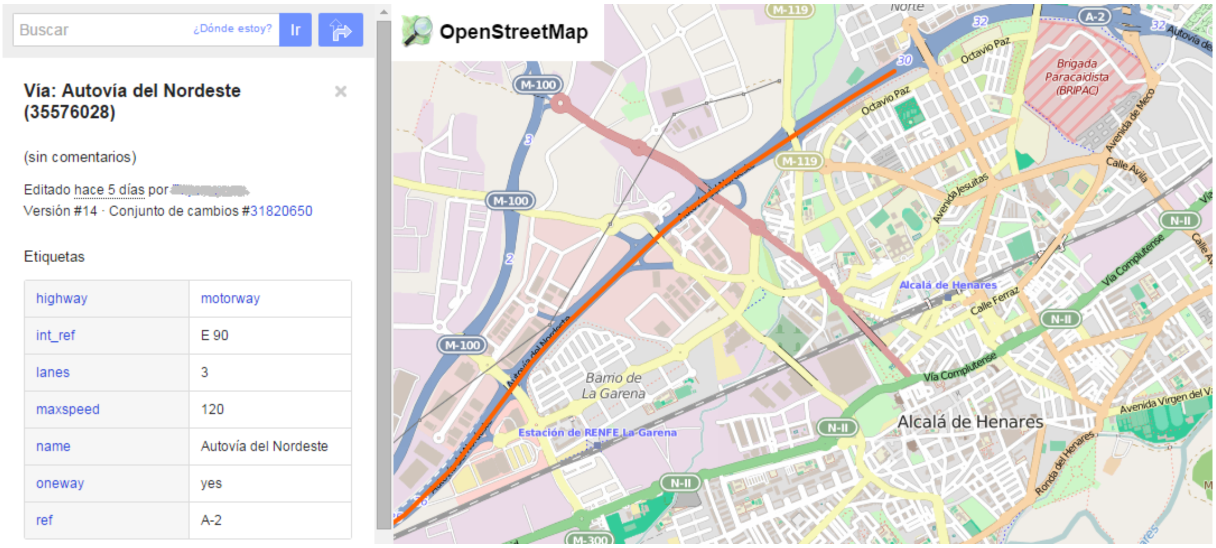


Figure 5.5: OpenStreetMap example of road information.

OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. As an open map created by the collaboration of several users and companies, the information it contains is not restricted to a simple set of roads and POIs (points of interest), but it contains very rich information about all of its elements (see Fig. 5.5). In the case of the roads, the additional information becomes very useful for evaluating the conditions in which a driver is driving on a certain road. The main indicators that can be extracted from OpenStreetMap and that are used currently in DriveSafe are:

- **Maximum allowed speed:** Maximum official allowed speed on a road. It can be used to calculate overspeeding, which may be an indicative of aggressiveness or drowsiness.
- **Lanes:** Number of lanes on a road. This, together with the detection of lane departures and some control over when the vehicle has incorporated to a certain motorway (i.e. previous position was in a motorway link), allows to estimate in which road lane is the vehicle positioned. This information might be useful for inferring aggressiveness (i.e. tendency to stay on the left lane of the road).
- **Area delimitation:** Several areas in OSM contain tags for differentiation of areas. For instance, areas such as a city are tagged as residential. Currently, DriveSafe is aimed for road use, so this feature is not used. However, it could be useful in the future to take into account the differences between driving in a city and a road.

OpenStreetMap is organized mainly as a set of nodes, ways and relations between them. Nodes indicate either POIs or serve as a structure to ways. Ways are used to describe any type of road or, in case of closed ways, to delimit areas. Ways can only be set between a set of nodes. Each node or way contains an undefined amount of tags that describe that element. For instance, in the road query shown in Fig. 5.5, a way describes a section of "Autovia del Nordeste", and its tags describe it as a motorway, with its official name, reference ID, and its official maxspeed and number of lanes for that section.

OSM has an online API that can be accessed via HTTP requests. By querying with a GPS position (lat and long), one can access the properties of nearby roads and thus obtain the desired information. Nevertheless, it has some limitations as it does not allow various petitions per second and it has a relatively

(a) (b)

Figure 5.6: Examples of overspeed detection and estimation of current lane. (a) Driver remains on right lane with a speed (107 km/h) lower than maximum speed (120 km/h), depicted with a green downwards arrow. (b) Driver is at the left lane with a speed (134 km/h) much higher than the officially allowed one (120 km/h), depicted with a red upwards arrow.

long delay on solving its queries, between 2 and 5 seconds. In order to solve this, the GPS position that DriveSafe uses to query OSM is the prediction of future position in that time interval, calculated as it was explained in Sec. 5.1.2.

Fig. 5.6 shows the new features that have been added to the interface. At the top of the view, the driver can see the maximum speed of the current road section, accessed via OSM. A green downwards arrow reflects whether the current speed is lower than the maximum allowed one, and a red upwards arrow reflects whether the current speed is higher than the allowed one. The other feature, the “current lane” detection, is shown as a number of road patches in the image that corresponds to the number of lanes that is given by OSM (3 lanes in both figures). A state machine is kept in DriveSafe to predict the current lane as a result of the detected lane departures and the information about the current and previous roads. For example, when the driver is in an incorporation to the motorway (reflected in OSM as motorway link) and he changes to the motorway, DriveSafe may infer that the current lane at that point will be the right-most of the road. The lane detection system of DriveSafe allows to detect whether a lane departure is performed to left or to right, which performs changes in the “current lane” state. The vehicle detection system is used as a correction of these states. For instance, if the state machine wrongly positions the vehicle in the left of all lanes and a vehicle is detected to overpass us from the left, then this position is corrected to be at least one more on the right. In the Fig. 5.6a the red car icon reflects the current position as the right-most of all lanes, and in Fig. 5.6b it is positioned in the left-most position.

5.2. Driver scoring

Scoring a driver with numeric values is essential for both simplifying things mathematically and encouraging good driving practices by awarding driving with a value that summarizes whether they drove in a good way. This section describes how all the indicators that were introduced in the previous section are gathered to form 7 independent scores and a total one. These scores represent: sudden accelerations, sudden brakings, sudden turnings, lane drifting, lane weaving, overspeeding and car-following.

5.2.1. Sudden accelerations, brakings and turnings

Accelerations, brakings and turnings are events that are produced due to either a distraction or impatience associated with aggressiveness. These events, detected by setting thresholds to the inertial sensors as was explained in Sec. 5.1.1, are used to establish their three associated scores.

The score is the result of taking into account the number and intensity of the events detected per Km through the Eq. 5.7, where (k_1 , k_2 , k_3) are constants experimentally calculated, and Eq. 5.8, where CDF_e represents cumulative distribution functions of the Gaussians for a normal driving, previously obtained for one of the users in the highway route.

$$Event\_km_e = \left[\left(k_1 \cdot Low_e + k_2 \cdot Medium_e + k_3 \cdot High_e \right) / Km \right] \tag{5.7}$$

$$Score_{event} = 1 - CDF_e \left(Event\_km_e \right) \tag{5.8}$$

$Score_{event}$ represents the score calculated for each of the events type e, which may be accelerations, brakings or turnings. Any of these events imply that there was a distraction or indication of aggressive behaviour. Therefore, the score is lower as higher is the amount of events produced. The constants k_1 , k_2 and k_3 have been set experimentally to 1, 4 and 8 respectively. This formula was tested in the previous DriveSafe work [7] and matched against a reference application that evaluates driving by means of the inertial events, AXA Drive [16].

5.2.2. Lane drifting

As it was described in Sec. 5.1.3.1, drifting is measured as the percentage of time that the driver remains inside a virtual lane of 1.2 meters around the centre of the lane. To produce the score, this percentage is calculated over a time window of 1 minute. For instance, if the driver keeps an entire minute driving correctly inside the virtual lane, the value of that window would be $\text{LANEX}_{60s} = 0{,}0$. On the other hand, if the driver keeps the whole minute outside of the virtual centre lane, the value for that window would be $\text{LANEX}_{60s} = 1{,}0$.

The final drifting score is calculated as the inverse of the mean of all the computed windows during the route plus their standard deviation:

$$SCORE_{drifting} = 1 - (\text{MEAN}_{\text{LANEX}} + \text{STD}_{\text{LANEX}}) \tag{5.9}$$

Remaining in the centre of the lane is the correct way of driving. Therefore, the score is higher as the lower are all the LANEX values, or the lower is the percentage of time spent outside the virtual lane.

5.2.3. Lane weaving

This score considers the relation between involuntary changes and voluntary:

$$SCORE_{weaving} = 1 - \frac{k_{invol} \cdot LD_{involuntary}}{LD_{voluntary} + LD_{involuntary}} \tag{5.10}$$

Involuntary lane departures are a negative fact because they imply drowsiness or inattention. Therefore, the score is lower as more lane departures have been produced involuntarily. k_{invol} is a factor that adjusts the punishment that is applied for each involuntary lane departure. By experimental tests, it has been set to $k_{invol} = 2$. Thus, each involuntary departure subtracts the score that is achieved by performing 2 departures correctly (voluntarily). The cause for this is that involuntary departures are not very common, so each one must have a higher impact on the score than performing one correctly.

5.2.4. Overspeeding

OSM (Sec. 5.1.5) is used to obtain information from the maximum allowed speed on the current road. Although this data is not available on all the existing roads, it is on the main ones, and it is being completed every day. The overspeeding value is calculated as follows, but only when there is maximum speed information available for the current road:

$$overspeed = \frac{speed}{maxspeed} \tag{5.11}$$

On the cases when there is no data available, it is not counted as a measurement, so it does not affect the overspeeding score. Otherwise, an accumulator is kept by adding the overspeed value only when it is higher than $th = 1{,}0$. On the cases that there is no overspeed ($overspeed \leq th$), nothing is added to the accumulator, but it is counted as a measurement. At the end, the accumulator is divided by the total number of measurements to produce the score:

$$SCORE_{overspeed} = 1 - \frac{accumulator_{overspeed}}{measureCount} \tag{5.12}$$

This way, the score remains high as the driver keeps under allowed maximum speed, but it is highly degraded as soon as the driver passes over that speed. In other words, keeping the speed is not rewarded (because it is an obligation) but passing the maximum speed supposes a “punishment” to the score, being higher as higher is the overspeed relation.

5.2.5. Car-following

In a similar way to what is done for overspeeding, car-following is evaluated as a score that is only degraded on the event of aggressive tailgatings. This is, the score remains high as long as the vehicle detection does not detect an ahead vehicle, or the one detected is not closer than 0.5 seconds of time of impact. Otherwise, on the cases when the driver is tailgating the ahead vehicle by keeping a time of impact under 0.5 seconds, the score suffers a high reduction. This considers the fact that keeping a safe distance is a driver’s obligation, while aggressive tailgating is a very dangerous behaviour that must be avoided.

A difference against the method followed for overspeeding is that also the cases when there is no vehicle detected are counted as a measurement that affects the score. This is due to the limitation of the vehicle detection module, which does not detect vehicles over 65-70 meters. Therefore, the lack of vehicle detection might also indicate that the driver is not tailgating a vehicle (by keeping a long distance), so it must be counted towards the mean score as a positive situation.

The “punishment” or points that are subtracted to the score are relative to the relevance of the tailgating:

$$Punishment = \begin{cases} 5, & \text{if } (timeImpact \leq 0{,}1s) \\ 4, & \text{if } (0{,}1s < timeImpact \leq 0{,}2s) \\ 3, & \text{if } (0{,}2s < timeImpact \leq 0{,}3s) \\ 2, & \text{if } (0{,}3s < timeImpact \leq 0{,}4s) \\ 1, & \text{if } (0{,}4s < timeImpact \leq 0{,}5s) \\ 0, & \text{otherwise} \end{cases} \tag{5.13}$$

Thus, the final car-following score is calculated as follows:

$$SCORE_{car-following} = 1 - \frac{SumPunishments}{Measures} \tag{5.14}$$

5.2.6. Global score computation

The global score is simply computed as the weighed sum of the 7 independent scores:

$$SCORE_{Global} = k_1 S_{accel} + k_2 S_{brake} + k_3 S_{turn} + k_4 S_{drift} + k_5 S_{weav} + k_6 S_{overspeed} + k_7 S_{carfollow} \tag{5.15}$$

Currently, each score is considered equally relevant:

$$k_1 = k_2 = k_3 = k_4 = k_5 = k_6 = k_7 = 1/7 \tag{5.16}$$

In future work, these constants could be rethought or weighed by means of higher-intelligence techniques, such as Fuzzy logic or other classification methods.

5.3. Behaviour modeling

Aside from the numeric scores, DriveSafe is able to infer degrees of certain behaviour types from the described indicators and some additional factors. While the original DriveSafe only distinguished drowsiness and distraction, the expansion in variables has allowed a wider range of behaviour profiles. Thus, DriveSafe now differs between three behaviours: drowsy, aggressive and normal driving, with an independent component of distraction that does not rely on any of the previous behaviours. This is due to the fact that distractions might be present even in a normal driving state, so they are treated as an independent event that may happen in all conditions.

The following sections describe how DriveSafe infers the ratios of the driving behaviour states by gathering the indicators that are relevant for each model and considering additional factors that were not used to produce the scores but give clues about the driver behaviour.

5.3.1. Drowsiness

Drowsiness is a behaviour produced by the lack of sleep or rest. During driving, it may have fatal consequences, as it produces a continuous lack of attention. It is reflected by the following indicators:

- **Lane weaving**: Involuntary lane changes are a relevant event that is produced in a drowsy state.
- **Lane drifting**: The inability to stay in the centre of the road lane unequivocally reflects drowsiness.
- **Additional component**: Frequency per km of Medium and High events (accel, brakes, turns) reflect if there is a prolonged state of inattention. Low events are not considered for drowsiness because they do not often imply a distraction, but movements within driving normality.

These indicators are considered as follows. The additional component or "Extra" is computed by considering the sums of all types of medium and high events, related to the amount of travelled kilometres by a factor of 2.

$$EXTRA_{drowsy} = 1 - \left(\frac{4 \cdot CountEvents_{medium} + 8 \cdot CountEvents_{high}}{2 \cdot distance_{km}} \right) \quad (5.17)$$

Finally, the ratio of drowsiness is weighed as follows. Weaving and Drifting have a higher factor due to their relevance and reliability against the extra component.

$$DROWSY_{ratio} = 1 - (0{,}4 \cdot SCORE_{weaving} + 0{,}4 \cdot SCORE_{drifting} + 0{,}2 \cdot EXTRA_{drowsy}) \quad (5.18)$$

5.3.2. Aggressiveness

Aggressiveness is the behaviour of being inclined to behave in an actively hostile way. During driving, aggressiveness is remarkably dangerous due to the lack of prudence in performing sudden events that are often taken to the limit of driver's control. The following indicators reflect this driving state:

- **Overspeeding**: The act of driving over the maximum allowed speed reflects impatience and thus aggressiveness.
- **Car-following**: The way the driver approaches or follows ahead vehicles implies aggressiveness if it is performed very closely (tailgating).
- **Additional components**:
  - Frequency of LOW events per km: Low events do not normally reflect a distraction but a sudden movement within normality. However, a high frequency of them per kilometre may reflect that the driver is behaving aggressively.
  - High amount of lane departures per Km: Although it does not imply a negative conduct, a high frequency of lane changes per kilometre may reflect impatience.
  - Keeping too much on left lane if the road has more than 2 lanes: On motorways that have more than 2 lanes, staying constantly in left lane (overtaking lane) reflects that the driver wants to drive faster than the other vehicles, and thus more aggressively. This component is only considered in motorways with more than 2 lanes.

The final aggressiveness ratio is computed as follows. Firstly, the amount of time spent on the left lane is computed similarly to the LANEX variable from lane drifting. Each second, the application computes a window of the last 1 minute that counts the percentage of time that is spent on the left-most lane if there are more than 2 lanes. The final score for "laneLeft" is computed considering the mean of all the windows computed during the route and their standard deviation:

$$SCORE_{laneLeft} = 1 - (meanWindows + stdWindows) \quad (5.19)$$

Considering this score and the other two additional components, the "EXTRA" component of aggressiveness is computed as follows:

$$EXTRA_{agressive} = 1 - \frac{\left(1 - \frac{CountLaneChanges}{2 \cdot distance_{km}}\right) + \left(1 - \frac{CountLowEvents}{2 \cdot distance_{km}}\right) + SCORE_{laneLeft}}{3} \quad (5.20)$$

Finally, the aggressiveness ratio is computed as follows. Overspeeding and car-following have a higher impact on the final ratio due to their higher relevance and reliability against the other three additional components.

$$AGGRESSIVE_{ratio} = 1 - \frac{(SCORE_{overspeed} + SCORE_{carfollow} + EXTRA_{aggressive})}{3} \quad (5.21)$$

5.3.3. Normal driving

The “Normal” ratio is computed directly as the lack of any other state:

$$NORMAL_{ratio} = 1 - (AGGRESSIVE_{ratio} + DROWSY_{ratio}) \quad (5.22)$$

Note that the ratio in Eq. 5.22 might become lower than 0, as the sum of agressiveness and drowsiness might be higher than 1. On this case, the final ratio is thresholded to 0. Thus, the ratio of normality cannot be higher than 0 as soon as other states are predominant.

5.3.4. Distractions

Distractions are not modelled as a general state but as singular events. A distraction is considered as a “one-time event” that may happen during any driving state, so it is modelled independently to the behaviour states of drowsiness, aggressiveness or normal, although the distraction frequency gives clues about those states. There is a distraction whenever there is any of the following events:

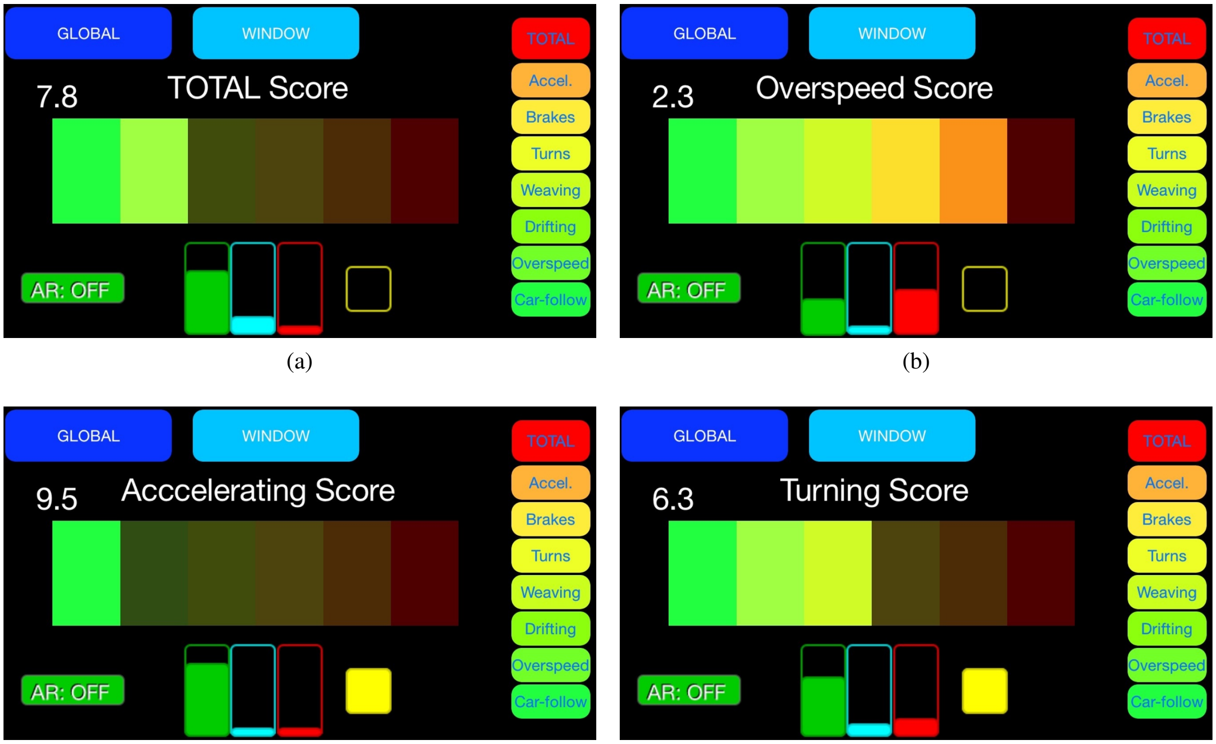
- Any Medium or High inertial event (accelerations, brakings, turnings).
- An involuntary lane departure.
- Time of impact to ahead vehicle becomes lower than 0.5 seconds.

Distractions are considered by DriveSafe in order to alert the driver on specific events during driving, but not to generate a individual score. However, as it was stated on the sections that describe the individual behaviour states, the frequency of specific distractions has been considered on the behaviour ratios as additional components.

5.4. Score GUI

The previous interface view without augmented reality from DriveSafe held a single inattention bar that was based on lane drifting. With the new expansion of indicators and behaviour modelling, this simplified interface has been changed to give complete feedback about all the scores and ratios. Thus, the single score bar has remained in the centre of the screen but now the user may choose what it shows. As Figs. 5.7 show, several buttons have been added to the view in order to personalize what the user can view in the large score bar.

The driver may choose to see either the score from accelerations, brakings, turnings, lane-weaving, lane-drifting, overspeeding, car-following, or the total score that considers all of them. Two additional buttons have been added to choose between the “global” scores, which shows the scores as the total until the current point during the route, or the “window” scores, which show each score as a result of only the last time window of 1 minute. While the global scores are also reflected at the end of the route,



(a) (b)

(c) (d)

Figure 5.7: Examples of scores shown in the new interface. a) View shows the Total current score (7.8); b) A Overspeeding score of 2.3; c) A Accelerating score of 9.5; d) A Turning score of 6.3.

the window ones may be of more interest during the route because they give instantaneous feedback about driver's actions. This way, a driver may receive an instantaneous degradation in the score when he performs a negative action (e.g. a sudden braking). This helps in encouraging good driving practices, as the driver receives constant feedback during driving of how his actions are considered in the score. Figs. 5.7 show 4 examples of this interface view. Fig. 5.7a shows a total score of 7.8, with the bar illuminated only on the first two green colours, while Fig. 5.7b shows a low overspeed score (2.3), with the bar illuminated until the orange level, indicating that the score is within a dangerous range.

Three bars have been also added at the bottom of the screen to indicate the ratios of normal (green), drowsy (light blue) and aggressive (red) behaviours. These can also be considered for either the global route or for a window of the last 1 minute. Figs. 5.7a, 5.7c and 5.7d show a predominance of the Normal ratio with low levels of drowsiness and aggressiveness, while Fig. 5.7b shows a predominance of the aggressive behaviour, which is mainly due to low scores in overspeeding and car-following (as it was seen in the previous section). A yellow square near the behaviour bars is activated when there has been a distraction in the last few seconds (as shown in Figs 5.7c and 5.7d).

5.5. Experiments

Collecting datasets to adequately evaluate DriveSafe is challenging. This is because dangerous driving events are not guaranteed to happen during routine driving experiences. Thus, we cannot accumulate enough examples of poor or dangerous driving to fully evaluate DriveSafe. Furthermore, it would be irresponsible to run an experiment that promoted risky behaviours. For these reasons, we evaluate DriveSafe using two different tests:

1. **Evaluation of Events**: Driving by testing a series of controlled vehicle maneuvers, where we safely "stage" dangerous driving events under controlled conditions, in which each driver is accompanied by a "co-pilot" who launches the controlled maneuver only when external conditions on the road are safe. The aim of these tests is to evaluate the correct detection of the different events in DriveSafe (the low level of analysis).

2. **Evaluation of behaviours**: The driver is told to perform routes by trying to emulate Normal, Drowsy and Aggressive behaviours, but without orders from the "co-pilot", who only takes notes. The aim of these tests is to evaluate the higher-level of inference in DriveSafe that evaluates the behaviour ratios.

5.5.1. Definition of test bed

The tests are performed on a variety of 4 drivers from our laboratory with 5 different vehicles.

Driver	Vehicle Model
D1	Audi Q5 (2014)
D2	Mercedes B180 (2013)
D2	Subaru Legacy AWD (1993)
D3	Citröen C4 (2006)
D4	Citröen C4 (2014)

Table 5.2: List of drivers and vehicles that performed the tests.

Each driver performs both types of tests. Firstly, a set of events are told to the pilot in order to perform them one by one whenever it is possible and safe in the road. They are described in Sec. 5.5.2. Secondly, each driver performs a route of 10-15 minutes by "simulating" a state (drowsy, aggressive, normal). In these routes, the pilot only receives main clues on how to simulate these states but does not receive any order during the route. These tests are described in Sec. 5.5.3.

The drivers are all men, one of them is aged around 45 years and has several years of driving expertise, and the other three are aged between 24 and 27 years, with between 3 and 6 years of driving expertise. About the vehicles used, the three first (Audi Q5, Mercedes B180 and Subaru Legacy) are vehicles with high engine torque, which affects the detection of inertial events performed by DriveSafe because their acceleration and turning movements are slightly more brusque. This will be seen in the following sections, as DriveSafe detected more of these events (specially accelerations) in these cars than on the other two (both Citröens), which are compact cars with lower engine cylinder capacity.

5.5.2. Evaluation of event/indicator detection and associated score

These tests have the aim of evaluating the right response of DriveSafe App to the performance of the actions that are collected by the driving indicators. Thus, the list of tests is performed 5 times (with all 4 drivers and 5 vehicles). The drivers are accompanied by a co-pilot that tells them to perform each action when it is safe in the road. The set of events and the amount performed of each one is listed below:

- **5 sudden accelerations**: Accelerate suddenly when speed is between 50 and 100 km/h.
- **5 sudden brakings**: Brake suddenly when speed is over 80 km/h and no vehicle is behind.
- **5 sudden turnings**: Turn suddenly when speed is over 80 km/h, within lane or overpassing a bit the contiguous lane when there is no other vehicle around.

- **10 lane departures (LD) to verify lane weaving**: Firtly perform 10 lane changes to calibrate the detector (they do not affect score) and then perform 10 lane departures at random times during driving, from which 5 are voluntary (normal lane changes) and 5 are involuntary (slowly as in a drowsy state).
- **2 x ∼30 seconds away from lane centre to verify lane drifting**: The driver is told to slightly avoid the centre of their current lane during 30 seconds as simulating not being perfectly aware of its position inside the lane. Duration might vary between 25 and 35 seconds.
- **2 x ∼30 seconds overspeeding**: The driver is told to keep a speed by at least 10 % more of the allowed speed during approximately 30 seconds (might vary between 25 and 35 seconds).
- **5 aggressive approaches to ahead vehicle**: The driver is told to approach a vehicle until a safe but aggressive distance (∼0.3 seconds of time of impact), remain a few seconds (5-10) and then go back or switch lane.

Table 5.3 shows the results of the event detection tests. The first column shows the amount of detected events/actions versus the total amount performed. The only ones that do not have a perfect detection rate are accelerations, that highly depend on the vehicle's engine and were difficult to reproduce in the slower ones; and Lane Departures (LD), for which the variance of the time in which the departures are performed is high in some drivers, and depends on more factors not considered such as traffic conditions.

The second column reflects the degradation in the windowed (last minute) score. The degradation of each event in the global score is not shown because it depends on route duration (i.e. the global score is more affected by a single event in the first kilometres of the route because less events have been performed overall), while the window of 1 minute is equal for all the scores. The degradation of the scores shown in this table does not reflect how the score is calculated but they just give an idea of the instantaneous feedback that the driver receives. For example, performing a high braking instantly passes the windowed braking score to 0.0 from 10.0, but that does not reflect how the global score will be, because it depends on more factors (e.g. distance ran, frequency of events, importance of each one).

Event	Det/Tot	Score (window) degrades by	Observations
Acceleration	20/ 25	∼0.1 (Low) / ∼6-7 (Med.) / 10 (High)	OBS1
Braking	25/ 25	∼0.1 (L) / ∼6-7 (M) / 10 (H)	OBS1
Turning	25/ 25	∼0.1 (L) / ∼6-7 (M) / 10 (H)	OBS1
Voluntary LD	28/ 25	Increase of ∼3.3 (out of 3 LDs per window)	OBS2
Involuntary LD	22/ 25	Decrease of ∼6.6 (out of 3 LDs per window)	OBS2
Lane-drifting (∼30s.)	10/ 10	Between ∼2.5 and ∼5 (dep. on duration)	OBS3
Overspeeding (∼30s.)	10/ 10	Between ∼5 and ∼7 (dep. on dur./speed)	OBS4
Aggressive Car-follow	25/ 25	Btw. ∼3 and ∼5 (dep. on dur./impact-time)	OBS5

Table 5.3: Results of event detection.

Below are described the individual observations about the tests and values that are shown in the table:

OBS1: The inertial events affect the score depending on their level. Low events do not affect remarkably because they are normal in an average driving. They are much more frequent than Medium and High events, which highly degrade the score when they are produced. The detection of inertial events works well. On the case of accelerations, it was difficult to force sudden accelerations because some test vehicles did not have enough engine torque to produce a remarkable acceleration on a speed over 50 km/h. On the case of brakings and turnings, the forced ones were perfectly detected (mostly as Medium events), and several others that are not reflected in the table and were produced during the driving operation were also detected (mostly Low events).

OBS2: The lane departures and their type is in general well detected. The false detections are due to the high differences in how the same driver performs some lane changes. One involuntary lane departure can degrade the score from 10 to 0 because not many departures are produced per minute and the score is relative to the total departures in the window. Therefore, the score in the window supposes only an instantaneous feedback, while the global score of the route supposes a more precise score because it keeps track of all the lane departures during the route and thus the estimated score depends relies on more data.

OBS3: Staying away from lane centre for a time intervals is perfectly detected and the score is well related to the time that the driver spends outside.

OBS4: Driving at a speed higher than the allowed one is perfectly detected while there is internet available in the phone to access OSM and the maxpeed parameter of the road is available in this API. This is normally available in the high motorways that were tested, becoming less available in secondary roads. The score seems well adjusted to the ratio of overspeed (speed/maxspeed) and to the duration of the overspeeding intervals.

OBS5: Aggressive car-following is well detected although the score is only affected when the ahead vehicle is at a time of impact of 0.5 seconds or less, considering non-aggressive the rest. The score is highly degraded if the tailgating is kept for more than 5-10 seconds, easily reaching scores of 0.0 in the window for longer tailgating intervals. It is also well related to the level (keeping 0.1 or 0.2 seconds of time of impact to ahead vehicle degrades score much faster than keeping 0.5).

5.5.3. Evaluation of driver profiling

These tests have the aim of evaluating the behaviour profiling (drowsiness, aggressiveness and normal driving). As it was explained in previous sections, the modelling of the behaviour is performed as three ratios that indicate an estimation of the predominance of each behaviour during driving. In order to evaluate these ratios, which have certain degree of subjectiveness, we have told the 4 drivers to try to emulate them without any additional indications during the route. Thus, each driver performs three different routes with predominance of each of the three behaviours in order to evaluate how DriveSafe assigns each of the ratios. The routes and simple indications that are given to each driver at the start of each one about how to emulate these behaviours is described below:

- **Drowsy:** On a trip emulating drowsiness, the driver is only encouraged to act slowly to events and be slightly unaware of the road lanes. Although the driver is not forced to perform any singular event, this translates into lane-drifting (as the driver is not consistent with his position on the lane), lane-weaving (as the driver does not change fluidly between lanes) and a few medium-high events such as brakings (as the driver response time gets slower and he reacts suddenly to dangerous events).
- **Aggressive:** On a trip emulating aggressiveness, the driver is only encouraged to drive fast, as being impatient. Without forcing the driver, this results in overspeeding, keeping low distances to ahead vehicle when it is not possible to overpass them, and a high density of brusque movements.
- **Normal driving:** The driver is told to drive as in their normal driving performance, even if it implies any degree of aggressiveness or drowsiness.

Below are presented the tables that gather the statistics collected for each of the drivers and routes. Table 5.4 shows the results for Driver 1, Table 5.5 for Driver 2, Table 5.6 for Driver 3 and Table 5.7 for Driver 4. These tables show simple information of each route as the durations and the time of day (T. of day) at which they were performed, and two sets of scores. The first is a set of subjective scores that each driver is asked in order to verify how they value their driving actions. Thus, each driver is asked a total score (Tot) that should reflect the average of scores, and a ratio for each behaviour: normal (Nor), drowsy (Drow) and aggressive (Agg) as how they think that they should be rated by an evaluation tool like DriveSafe. The second set of scores shown in the table shows the main values given by DriveSafe App for the same concepts (total score and behaviour ratios).

State	T. of day	Duration		Driver's Subjective Score				DriveSafe's Score			
		Time	Km	Tot	Nor	Drow	Agg	Tot	Nor	Drow	Agg
Drowsy	19:50	10 m.	13	6.5	3-4	7-8	1	7.4	3.0	5.7	1.3
Aggressive	19:41	8 m.	13	5.5	3-4	1	8	7.0	4.4	1.1	4.5
Normal	19:31	8 m.	12	8	7-8	1	2	9.5	6.9	1.5	1.6

Table 5.4: Results of Behaviour-simulated routes for Driver 1 (Audi Q5).

State	T. of day	Duration		Driver's Subjective Score				DriveSafe's Score			
		Time	Km	Tot	Nor	Drow	Agg	Tot	Nor	Drow	Agg
Drowsy	17:00	15 m.	17	6-7	4	6	0	7	3.3	4.9	1.8
Aggressive	17:16	13 m.	18	5	4	1	7	7.7	3.1	1.3	5.6
Normal	17:30	12 m.	14	9	8-8.5	1	1	9.2	6.8	1.0	2.2

Table 5.5: Results of Behaviour-simulated routes for Driver 2 (Subaru Legacy - 1993).

State	T. of day	Duration		Driver's Subjective Score				DriveSafe's Score			
		Time	Km	Tot	Nor	Drow	Agg	Tot	Nor	Drow	Agg
Drowsy	12:36	6 m.	11	5-6	4	7-8	0-1	8.2	3.5	4.6	1.9
Aggressive	12:52	6 m.	13	5	4-5	1-2	7-8	7.9	3.7	1.6	4.7
Normal	12:44	8 m.	12	8.5-9	8.5-9	1	1.5-2	9.5	7.7	0.8	1.5

Table 5.6: Results of Behaviour-simulated routes for Driver 3 (Citröen C4 -2006).

State	T. of day	Duration		Driver's Subjective Score				DriveSafe's Score			
		Time	Km	Tot	Nor	Drow	Agg	Tot	Nor	Drow	Agg
Drowsy	16:42	5 m.	8	4	4	7	1-2	7	2.9	5.9	1.2
Aggressive	16:56	7 m.	13	4	4	1	8-9	8.3	4.3	1.7	4.0
Normal	16:47	6 m.	10	8	8	2	3	8.8	5.6	3.0	1.4

Table 5.7: Results of Behaviour-simulated routes for Driver 4 (Citröen C4 -2014).

While DriveSafe tries to analyse an objective ratio for each behaviour state, the values given by the Drivers are subjective and depend on their conception on how dangerous are their actions. Considering this, it is interesting the fact that a driver that drove the least aggressively in the aggressive route (Driver 4) is the one that thought that he should be "punished" the higher score of aggressiveness (8-9). One could think that DriveSafe is not well calibrated and that if he thinks that he performed very aggressively then he should have a higher aggressive ratio. A fact is, that this driver drove subjectively less aggressive in the eyes of the co-pilot that supervised all the drivers. He performed less overtakings, drove with a lower relative speed, and approached the vehicles keeping a longer distance than the other drivers in their aggressive route. Still, he perceived that he should have almost the maximum aggressiveness ratio (8-9). This is due to the fact that each driver considers certain levels as absolutely excessive. The few tailgatings and overspeeding intervals that this driver performed were perceived by him as something very dangerous, while some other driver could consider tailgating and overspeeding as something normal in their driving operation.

The scores and ratios provided by DriveSafe for these drivers and routes seem in general well calibrated. These routes have been performed within several safety standards, so the fact that the aggressive or drowsy ratio in their corresponding route is within 4 and 6, and not higher, seems appropriate, because these behaviours were simulated under normal conditions and without imminent danger to the vehicle's occupants. Therefore, the fact that these ratios are not higher on the simulations does not imply that a driver in a real environment cannot obtain higher values, as he could drive in specific cases with less awareness than the safety measures taken in the tests. Aside the specific value ranges of each score and ratio, the predominance of the ratio assigned by DriveSafe to each behaviour in the tests is clear in each route, demonstrating that the application responds correctly to the stimuli. The value that perhaps suffers less correlation with the routes is the Total score, as one can notice that it is around 7-8.5 for the behaviour-simulated routes, and only over 8.5 for the normal ones. This is due to the fact that it is a mean of 7 variables, so all of them would have to be low for the score to degrade under 7, and it is difficult for a driver to perform badly in all indicators. However, this does not suppose an incongruence in the scores, as it has less relevance to talk about a total score that is just a mean of 7 independent and dissimilar values instead of the other 3 behaviour ratios, which seem well adjusted and already include the independent scores.

The Table 5.8 shows again the behaviour ratios together with each of the individual scores given by DriveSafe, sorted by behaviour type of the route and by each driver.

State	Driver	SCORES								RATIOS		
		Tot	Acc	Bra	Tur	Weav	Drift	Overs	Carfoll	Nor	Drow	Agg
Aggres.	D1	7.0	9.7	9.8	9.6	10	7.9	**4.8**	6.3	4.4	1.1	**4.5**
	D2	7.7	9.9	9.4	9.7	9.5	7.3	**3.3**	4.8	3.1	1.3	**5.6**
	D3	7.9	10	10	9.8	10	5.9	**4.1**	5.4	3.7	1.6	**4.7**
	D4	8.3	10	9.8	9.9	7.3	8.2	**4.9**	8.1	**4.3**	1.7	4.0
Drowsy	D1	7.4	10	6.7	9.8	**2.7**	4.2	8.7	9.5	3.0	**5.7**	1.3
	D2	7.0	10	**3.7**	9.5	3.9	4.0	7.8	10	3.3	**4.9**	1.8
	D3	8.2	10	9.9	9.9	**4.0**	4.6	9.4	9.7	3.5	**4.6**	1.9
	D4	7.0	10	3.9	9.9	**0.0**	5.4	10	10	2.9	**5.9**	1.2
Normal	D1	9.2	9.9	9.9	9.5	9.1	**8.3**	8.6	9.5	**6.9**	1.5	1.6
	D2	9.2	9.8	9.5	9.4	10	8.5	**8.4**	9.1	**6.8**	1.0	2.2
	D3	9.5	10	9.9	9.9	10	**8.1**	9.0	9.7	**7.7**	0.8	1.5
	D4	8.8	9.9	9.9	10	**3.9**	8.6	9.6	9.9	**5.6**	3.0	1.4

Table 5.8: Complete list of DriveSafe scores per behaviour state and driver. In the case of the scores, the minimum of the 7 variables is highlighted for each driver and route. In the case of the behaviour ratios, the predominant (maximum) one for each driver and route is highlighted.

The scores of inertial events (accelerations, brakings, turnings) are normally high due to the fact that the common events that do not suppose danger are chategorized as Low and they do not affect remarkably the score. Thus, the score for inertial events is only reduced with Medium and High events, which depend on road situations and are not likely to happen on short intervals such as the performed test routes (5-15 minutes). The rest of the individual scores seem appropriate, as the drowsy simulations produce lower scores in weaving and drifting, and the aggressive ones produce lower scores in overspeeding and car-following, which are the ones considered as more relevant to their respective ratios.

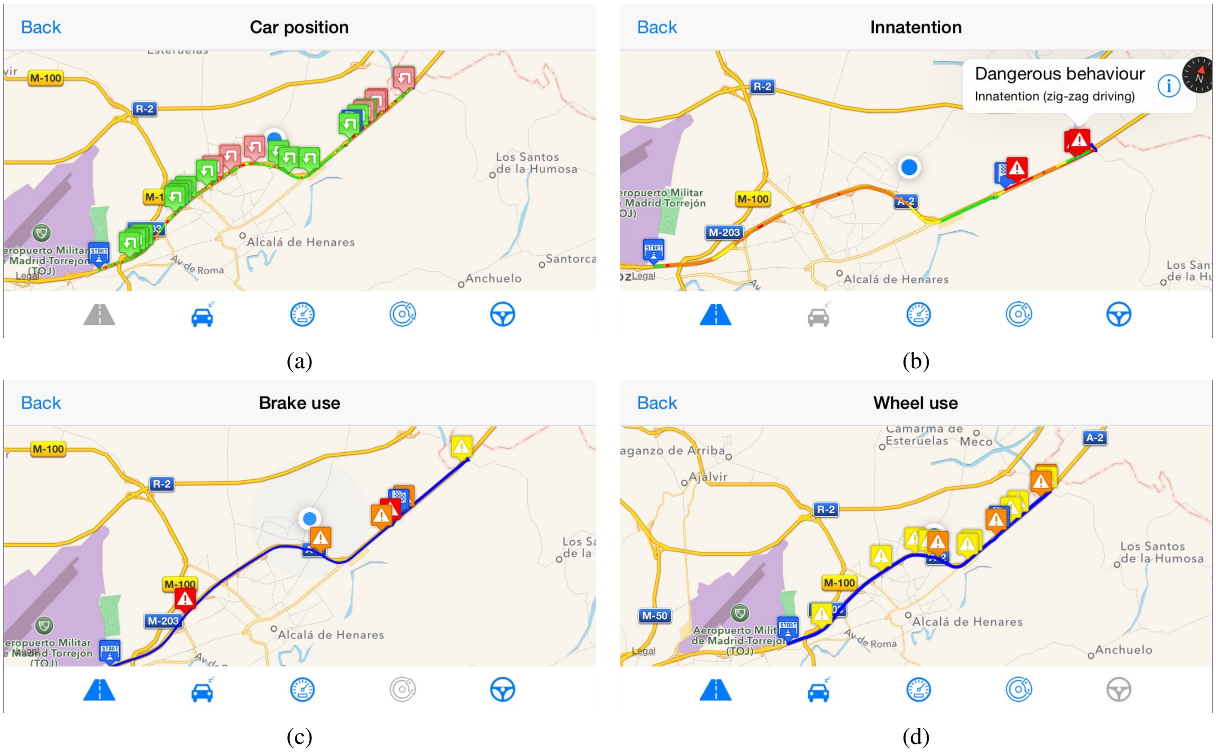


Figure 5.8: Examples of the map interfaces that collect some events produced during one of the test routes. a) The Car position map indicates information such as the performed lane changes, voluntary (green) or involuntary (red). b) The Inattention map shows with colours the sections where the lane-drifting was higher and with red exclamations the points where it was critical. c) The Brake use map indicates the braking events (yellow for low, orange for medium and red for high). d) The Wheel use map indicates the turning events, with the same colours as the Brake map.

5.5.4. Qualitative evaluation

It is difficult to evaluate the good functioning of DriveSafe as driving evaluation tool due to the subjectiveness in the behaviours and in the relevance of certain events. Also due to the difficulties in executing tests that evaluate dangerous situations, mainly due to the fact that performing them for tests supposes a risk itself. For instance, events such as a sudden turning are only categorized as “High” if they exceed a very dangerous threshold that may suppose losing traction in several cases, so trying to obtain a “High turning” event detection for testing is not a safe practice.

For these reasons, aside from the previous quantitative tests, we present a general idea on how each of the parameters that DriveSafe evaluates worked during the different tests:

- **Accelerations**: They are well captured mainly for the low cases, but it is difficult to produce a high or medium acceleration on a motorway due to the high speeds. However, accelerations do not normally suppose a dangerous thing so the detection of these events is not strongly relevant. During the tests, only the powerful vehicles (i.e. Audi Q5, Mercedes and Subaru) could achieve a medium-high acceleration performed over a speed of 50km/h.

- **Brakings**: Brakings are well captured on all cases. Low and medium brakings were correctly detected during the forced tests. High brakings were barely achieved during the tests due to the high threshold that is imposed to detect a high one, but they were mostly detected on extreme braking cases that were not forced (i.e. a vehicle suddenly positioning in front of the driver and thus obliging to perform sudden braking), which led us to think that this detection is well calibrated.

- **Turnings**: Turnings are also well captured on all cases. Similarly to the case of brakings, low and medium turnings were correctly evaluated during the forced tests, while high turnings were mostly achieved on extreme cases that were not being demanded by the co-pilot but happened as an unforeseen occurrence during the multiple driving hours taken for testing.

- **Weaving**: Performs well within its definition, although it is very subjective to evaluate whether a change is involuntary or voluntary. Slow changes can be produced due to high traffic (e.g. if driver is trying to gain some space on the adjacent lane by changing slowly), which might cause some false positives during a normal driving routine. However, although the driver could consider that specific cases detected by DriveSafe are incorrect (e.g. one change detected as involuntary when it was voluntary), the overall weigh of the lane changes performed slowly (involuntary) versus the ones that are done fast (voluntary) seems to affect correctly the ratio that infers drowsiness.

- **Drifting**: Performs well and reflects correctly the state of inattention. It has high reliance on the lane detection, which may suffer some size changes due to shadows or incorrect lane paintings on the ground. However, in general, it works robustly and the score obtained seems adjusted to what was perceived during the tests.

- **Overspeeding**: It has high reliance on OpenStreetMap, so it depends on the available internet connection and the availability of information for each type of road in the API. In general, the coverage on the main motorways is excellent so the overspeeding factor works well. In the case of secondary roads it depends on the region. However, this will be solved in the near future due to the exponential growth of the OSM data. This is also a singular problem in some countries like the one used to perform the tests (Spain), for which OSM data is less complete than for others in the European Union (e.g. Germany).

- **Car-following**: It performs well despite its reliance on the vehicle detection module. This is due to the fact that the vehicle detector recall rate is high and it is close to 100 % on the relevant intervals that are considered for car-following (time of impact lower than 0.5 seconds). The punishment in the score associated to the detected tailgatings seems appropriate, although it could be improved in future works by adding complexity to the analysis of how the driver approaches each vehicle.

And a qualitative idea of how the different behaviours were inferred:

- **Drowsy and Aggressive ratios:** This ratio seems well related to the available indicators on DriveSafe. As explained in the previous section, the fact that the tests could not achieve a ratio higher than 6 is justified, and higher ratios are reserved to those that drive truly in a drowsy or aggressive state that supposes a danger to other drivers. However, aside from the value ranges, the tests demonstrated that DriveSafe correctly detects the predominance of the specific behaviours.

- **Normal ratio:** This ratio depends directly on the lack of the others, so it relies on the correct detection of aggressiveness and drowsiness. The values that resulted on the tests seem appropriate, as this ratio was predominant as far as no dangerous behaviours were detected.

Chapter 6

Conclusions

The development and spread of the systems that perform driving analysis remarkably depends on how the driver is allowed to interact with the system and understand its use as a positive thing. The known concept of “Pay-As-You-Drive”, which tries to make the cost of services for each user dependant on how efficiently they make use of them, is expandable to either governments that could set lower taxes to those citizens that produce less contamination by moving more efficiently, or insurance companies that could set the rules for payment of insurances depending on how each person drives. On the case of insurers, this territory has been explored in the last years with the installation of “Black-boxes” that evaluate driving. However, this can be seen as an intrusive approach by the vast majority of drivers, which do not have any control of these systems. The rapid expansion of the smartphones as sensing platforms favoured the exploration of new ways to evaluate driving with the consent of the user, that normally carries his phone during the day and its use does not suppose an additional load of time and money.

DriveSafe App was able to seize this niche and present itself as a good alternative to black-boxes by evaluating and scoring driving in a robust and user-friendly way. The present work does not expect to reinvent DriveSafe, which involves various years of work, but it aims to expand its features and its analysis by exploiting all the available capabilities of a smartphone. Some of them have not been explored in the related works due to the complexity of using features like vision, thus producing that most works on driving analysis are mainly based on inertial sensors and GPS data. DriveSafe App is now presented as one of the most complete alternatives in the state of the art to perform driving analysis. It is a tool that is highly useful for both: drivers, that seek to improve all aspects of their driving in order to make it more efficient and safe; and for institutions and companies, which can make good use of massive driving data in order to improve conditions on the driving environment or make the collection of service payments more fair and dependant on the user.

This thesis has focused on solving three main problems. Firstly, the communication capabilities of the application have been solved by the development of a server that manages the data uploaded by a client inside DriveSafe. Secondly, an ahead vehicle detection and tracking module has been developed and integrated in the application in order to enhance its perception of the dynamical entities that coexist on the road and affect driving. Finally, new variables and indicators have been extracted of the available smartphone capabilities, including connection with OSM, to rearrange the driver analysis performed by DriveSafe.

6.1. Main contributions

The main contributions of this thesis that have been presented in previous chapters are summarized below:

- **Data server**: Managing data in an unified way is a necessity in DriveSafe due to the fact that it is thought for massive use. For this reason, a server based on LAMP (Linux, Apache, MySQL, PHP) has been designed and programmed to interact with a client module in the application that uploads the data collected by the drivers. The process is now transparent to the user and the information is now organized correctly in a joint place, which facilitates the analysis of massive data.

- **Ahead vehicle detection and tracking system for Smartphones**: Vehicle detection has several applications in the computer vision field and in the intelligent transport systems community. In this work, we present an approach that considers the road geometry and uses multi-scaling techniques in order to reduce the high associated computational costs and make it viable for smartphones. The experimental results, evaluated on a publicly known motorway dataset, prove similar to other state-of-the-art works despite of the smartphone computational constraints. Its integration in DriveSafe App allows a richer driver analysis that now considers ahead vehicles on the road as an important part on the driver's actions. This factor is omitted in most driver analysis platforms presented in the literature due to the implementation complexity.

- **Driver analysis**: New variables for driver analysis have been introduced in DriveSafe based on new features such as the ahead vehicles and the road information provided by online map APIs like OpenStreetMap. The previous variables have been rearranged together with the new ones to produce a wide set of indicators from which DriveSafe exports 7 independent scores and a rich behaviour analysis based on a series of ratios for each of the main states that affect driving: aggressiveness, drowsiness and normal driving. Varied tests have been carried out to demonstrate that the detection of indicators and the analysis performed by the application is correct and that it responds correctly to the stimuli by showing a correct predominance of the behaviour ratios on different simulated states.

6.2. Future work

Future work will involve both performing technical improvements in the application and further development of the theoretical concepts applied to driver analysis. On the technical case, this comprises the launch of the application on the market, adaptation to other smartphone types, improvements on the user interface and adjustment of parameters. On the theoretical case, this could involve developing techniques for massive data analysis to the data collected in the server or adjusting the theoretical values that weigh each indicator by considering more complex parameters (such as the profile of the driver, or the type of vehicle). The specific ideas about what will comprise future work related to this thesis are summarized below:

- **Data server**: The server developed for DriveSafe will allow the application of complex techniques for massive driver analysis with the aim of facilitating the extraction of rich information if the application becomes widespread. The globalization and exponential increase in the data that is available in the society has promoted the concept of Big Data, a term referred to data sets that are so large or complex that traditional data processing applications are inadequate. Thus, complex techniques are being developed in the research community to extract patterns from vast amounts of data in efficient ways. The application of these techniques to the data collected by DriveSafe would facilitate the extraction of relevant statistics about driving behaviours, which is of interest to governments and companies.

- **Ahead vehicle detection and tracking system**: Regarding vehicle detection, some complex techniques could be further researched to improve robustness against lightning effects like shadowing, or improve detection of other types of vehicles that have a pattern that is remarkably different to cars, such as trucks and motorbikes. Although the presented approach succeeds in reducing the decay on detection performance associated with the computational costs, the future release of smartphones with more processing capabilities (e.g. iPhone 7) will allow a wider margin for improving the detection rates on the far distances by using higher resolutions.

- **Driver analysis**: The driver analysis could be improved by adding more variables to the ones presented in this work or by making more complex the use of the ones that are currently used. On the first case, higher integration with the vehicle by means of Bluetooth or an OBD-II connector could be used to extract objective vehicle parameters and infer a consumption model of the driver. On the second case, the thresholds and factors used to extract and weigh the different indicators that comprehend driver analysis could be adapted or make dependant on other complex factors like vehicle or driver types. This could be performed by the integration of Fuzzy logic, like it was proposed in the work in [33], and more complex calibration processes to adapt the values correctly.

The new version of DriveSafe App will be uploaded to the application market in the near future, becoming available for all users. Its high modularity would allow also the adaptation to versions that better suit the needs of specific companies, such as transport companies that need to evaluate specific behaviours in their truck fleet. The techniques developed in this project are not constrained to driver analysis, but are exportable to a wide set of applications such as safety. Some parts of DriveSafe (e.g. lane or vehicle detection modules) are directly related with the Advanced Driver Assistance Systems (ADAS) that are being currently developed by vehicle manufacturers to integrate complex safety systems on the new transport systems, and the foundations are the same that the needed for the implementation of self-driving vehicles but using better sensors and more processing capabilities.

Bibliography

[1] *Road safety in the European Union: Trends, statistics and main challenges*. European Comission, March 2015. [Online]. Available: http://ec.europa.eu/transport/road_safety/pdf/vademecum_2015.pdf

[2] N. H. T. S. Administration *et al.*, "National motor vehicle crash causation survey (dot hs 811 059)," *Washington, DC*, 2008.

[3] Mobileye applications, http://www.mobileye.com/technology/applications/.

[4] Autovue lane departure warning, http://www.iteris.com/upload/datasheets/LDW_Final_web.pdf.

[5] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: insights from a controlled field study," in *Proceedings of the 11th Int. Conf. on Mobile and Ubiquitous Multimedia*. ACM, 2012, p. 36.

[6] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "Pripayd: Privacy-friendly pay-as-you-drive insurance," *IEEE Trans. on Dependable and Secure Computing*, vol. 8, no. 5, pp. 742–755, 2011.

[7] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes, and R. Arroyo, "Drivesafe: an app for alerting inattentive drivers and scoring driving behaviors," in *IEEE Intelligent Vehicles Symp. (IV)*, 2014, pp. 240–245.

[8] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 596–614, 2011.

[9] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 127–140.

[10] Smartphone App. iOnRoad, http://www.ionroad.com/.

[11] Augmented Driving, https://itunes.apple.com/tw/app/augmented-driving/id366841514?mt=8.

[12] M. Lan, M. Rofouei, S. Soatto, and M. Sarrafzadeh, "Smartldws: A robust and scalable lane departure warning system for the smartphones," in *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conf. on*, 2009, pp. 1–6.

[13] C.-W. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de Oca, Y. Cheng, M. Lin, L. Torresani, *et al.*, "Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 13–26.

[14] Aviva PLC. Aviva RateMyDrive, http://www.aviva.co.uk/drive/.

[15] State Farm Mutual Automobile Insurance Company. (2013). State Farm driver feedback, https://www.statefarm.com/customer-care/download-mobile-apps/driver-feedback.

[16] AXA Drive, https://itunes.apple.com/es/app/axa-drive/id716514409?mt=8.

[17] Greenroad, http://greenroad.com/.

[18] D. Johnson, M. M. Trivedi, *et al.*, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conf. on*, 2011, pp. 1609–1615.

[19] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 234–239.

[20] M. Van Ly, S. Martin, and M. M. Trivedi, "Driver classification and driving style recognition using inertial sensors," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 1040–1045.

[21] R. Araújo, Â. Igreja, R. De Castro, and R. E. Araujo, "Driving coach: A smartphone application to evaluate driving efficient patterns," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 1005–1010.

[22] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 1, pp. 91–102, 2015.

[23] A. B. Rodríguez González, M. R. Wilby, J. J. Vinagre Díaz, and C. Sánchez Ávila, "Modeling and detecting aggressiveness from driving signals," *Intelligent Transportation Systems, IEEE Trans. on*, vol. 15, no. 4, pp. 1419–1428, 2014.

[24] P. Angkititrakul, T. Ryuta, T. Wakita, K. Takeda, C. Miyajima, and T. Suzuki, "Evaluation of driver-behavior models in real-world car-following task," in *Vehicular Electronics and Safety (ICVES), 2009 IEEE International Conference on*, 2009, pp. 113–118.

[25] L. M. Bergasa, J. Nuevo, M. Sotelo, R. Barea, M. E. Lopez, *et al.*, "Real-time system for monitoring driver vigilance," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, 2006.

[26] I. G. Daza, L. M. Bergasa, S. Bronte, J. J. Yebes, J. Almazán, and R. Arroyo, "Fusion of optimized indicators from advanced driver assistance systems (adas) for driver drowsiness detection," *Sensors*, vol. 14, no. 1, pp. 1106–1131, 2014.

[27] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Trans. on Pattern Analysis & Machine Intelligence*, no. 2, pp. 199–213, 1992.

[28] J. C. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.

[29] S. Sivaraman and M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.

[30] C. Caraffi, T. Vojir, J. Trefny, J. Sochman, and J. Matası, "A system for real-time detection and tracking of vehicles from a single car-mounted camera," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2012, pp. 975–982.

[31] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. 511–518.

[32] M. Gabb, O. Lohlein, R. Wagner, A. Westenberger, M. Fritzsche, and K. Dietmayer, "High-performance on-road vehicle detection in monocular images," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2013, pp. 336–341.

[33] L. Castangia, P. Grisleri, P. Medici, A. Prioletti, and A. Signifredi, "A coarse-to-fine vehicle detector running in real-time," in *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2014, pp. 691–696.

[34] A. Haselhoff and A. Kummert, "A vehicle detection system based on haar and triangle features," in *IEEE Intelligent Vehicles Symp. (IV)*, 2009, pp. 261–266.

[35] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[36] S. Sivaraman and M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267–276, 2010.

[37] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 906–917, 2013.

[38] F. Garcia, A. Prioletti, P. Cerri, A. Broggi, A. de la Escalera, and J. Armingol, "Visual feature tracking based on phd filter for vehicle detection," in *Int. Conf. on Inf. Fusion (FUSION)*, 2014, pp. 1–6.

[39] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, pp. 1–10, 2001.

[40] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[41] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[42] J. Arróspide, L. Salgado, and M. Nieto, "Video analysis-based vehicle detection and tracking using an mcmc sampling framework," *EURASIP Journal on Advances in Signal Processing*, no. 1, pp. 1–20, 2012.

[43] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[44] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2003, pp. 264–271.

[45] M. Marszatek and C. Schmid, "Accurate object localization with shape masks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[46] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards life-long visual localization using an efficient matching of binary sequences from images," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[47] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 41–54.

[48] Abstand und Abstandsvergehen [Distance and Distance Offense] (in German). Retrieved 15 June 2015. https://www.bussgeldkatalog.org/abstand/?gclid=CO2ckuGVycMCFanMtAodpwkAcw.