Bachelor Thesis

Geometric models for video surveillance in road environments: vehicle tailgating detection

Eduardo Pla Sacristán

Tutor: Iván González Díaz

September 2016

Bachelor's Degree in Audiovisual System Engineering
Universidad Carlos III de Madrid

A mi madre

"Hey you, don't tell me there's no hope at all
Together we stand, divided we fall"
- Pink Floyd

Abstract

Traffic accidents constitute one of the main causes of death in many countries. Despite the current efforts devoted to mitigate the effects of road incidents, there are still some variables affecting this problem which are not yet under control or regulation. Spain, for instance, still lacks official regulations about especially risky driving behaviours, such as tailgating. In many cases, the rationale behind is that these behaviours are hard or expensive to detect reliably, thus limiting the extent of the automatic detection systems.

This paper proposes a method to identify certain elements in road scenarios, define geometric models that allow computing quantitative measures of the scene and, consequently, detect offending driving behaviours. In this work, we have focused on the particular case of study of tailgating detection. However, the proposed geometric models might become the basis of many other useful applications.

Index Terms — Computer vision, video analytics, traffic surveillance, geometric models, tailgating detection.

Compact Index

Table of contents

List of figures

List of tables

List of acronyms

Subsequently, a list is presented containing the relevant acronyms used in this manuscript. The first field of the table represents the acronym. The second field states its meaning and the third and final field points out the section of the manuscript where the term is first mentioned.

Acronym	Meaning	Section
OCR	Optical Character Recognition	2.1.2
SVM	Support Vector Machine	2.2.1.1
KOAD	Kernel-based Online Anomaly Detection	2.2.1.2
OCNM	One-Class Neighbour Machine	2.2.1.2
H	Homography transformation matrix	2.3.1
DLT	Direct Linear Transformation	2.3.1
SVD	Singular Value Decomposition	2.3.1
RANSAC	Random Sample Consensus	2.3.2
GMRE	Geometric Models for Road Environments	3.2
DVID	Driving Violation Infraction Detection	3.2
DL	Discontinuous Line	3.2
BL	Border Line	3.2
DA	Detection Area	3.2
SPM	Six-point method	3.3.4

1 Introduction

Since the widespread growth of automobile use worldwide during the XX century, road safety has become a key issue for modern societies. Many strategies have been proposed to address this problem, such us regulated driving licenses, restrictive signalling and, more recently, speed monitoring through radar systems. The advances of technology have allowed traffic regulation entities to incorporate new preventive systems to real scenarios. Among the broad family of technologies applied to road environments, computer vision systems have emerged as interesting solutions for certain problems due to their reduced cost and easy deployment. However, despite the incessant efforts to improve road safety, there are still some areas in which the control is little or even non-existent. The presented project attempts to solve one of these problems: tailgating.

Tailgating is a driving behaviour defined to occur when a vehicle drives behind another at a dangerous distance, too short to avoid a crash in the event of a sudden stop of the vehicle in front. This particular driving behaviour clearly represents a threat regarding road safety, and yet, as it will be shown in this report, it is not trivial to monitor or regulate it. Within urban areas, this kind of behaviour may result in minor accidents (Fig. 1). However, when the speed at which the vehicles are circulating is higher, for instance, in highways, collisions may result in fatal accidents, and tailgating becomes exponentially more dangerous. Via this manuscript, we will explain our attempt to tackle this problem from the computer vision angle.

Figure 1. Collision caused by tailgating.

This opening chapter presents the general overview of the presented project, along with a brief section describing the organization of this manuscript. In addition, the current socio-economic environment will be presented in order to properly introduce the area of research that concerns this project. After that, we will discuss the context of the investigation, describing the particular scenario in which the project is developed. Finally, we will explain the motivation and goals that led to its development and conclude with the work methodology that was used to approach the problem.

1.1 General overview

This project is the result of the collaboration of the author, Eduardo Pla Sacristán, with the Signal Theory and Communications department at the Carlos III University. The collaboration that concerns this project lasted from September 2015 to July 2016, and it was directly supervised by the Multimedia Processing Group, and more specifically by Ivan González Díaz and Fernando Díaz de María. In this manuscript, the author presents a system that could improve road safety at certain levels that are not under current regulation. In particular, he attempts to detect tailgating behaviours in the analysed scenes. Hopefully, this work will lead to further investigations to prevent potential accidents.

To that end, this project first aims to define geometric models that help defining certain driving behaviours in a video flow. In particular, traffic scenes are analysed, with the final goal of tailgating detection. For this, a computer vision system will be developed.

The overall system will be divided into two different blocks or subsystems, each oriented to evaluate and process the provided input in different stages. The first subsystem will be devoted to define de geometric models that define the road, and the second subsystem will be used to process the video flow using these geometric models, obtaining as an output the desired behaviour detection.

1.2 Socio-economic environment

Traffic related projects have a well-established socio-economic environment, as this is an area of research that goes back a long time ago. One of the most important social aspects regarding means of transportation is safety, as it determines the amount of users it will have. For instance, regarding aviation, trends clearly show positive advances in safety, as it is shown in Fig. 2.

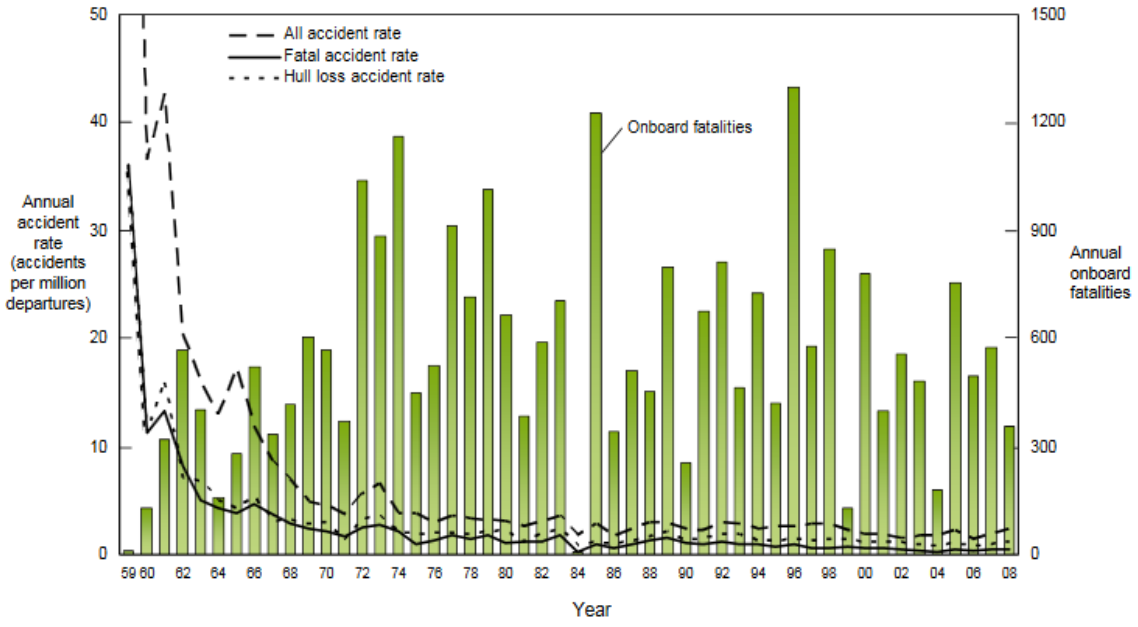


Figure 2. Evolution of aviation accidents since 1960. Figure taken from [2].

Moreover, statistics regarding commercial jet airplanes show that with this safety increase, a dramatic increase in number of passengers accompanied [2]. Regarding our particular application, which concerns road traffic environments, a similar scenario can be observed. Since the beginning of the XX century, the evolution of annual deaths per distance travelled has kept an inverse proportion with the total number of distance travelled by vehicles [3]. According to the commented statistics, we can affirm that in order for the automobile industry to keep growing, not only socially but also economically, road safety needs to keep improving. Therefore, the area of research in which this project is involved could become a key to the future of traffic safety.

1.3 Context

The general scenario that encompasses this project includes all aspects of road safety. Administrations usually rely on detection techniques such as radar to monitor traffic. Additionally, human resources are used in some cases, like preventive alcohol detection units, which randomly test drivers in order to check the level of alcoholism. The usual regulation method used to prevent accidents is sanctioning, although sometimes awareness-raising campaigns are also launched [5].

In this project, however, the main angle from which the problem is tackled is Computer Vision. This area of research has focused mainly on anomaly detection algorithms regarding traffic safety, but when looking for precedents referred to a specific driving behaviour like tailgating, there is a dramatic lack of research literature. Nonetheless, there are certain lines of research that represent an interest to this project's development, which will be later discussed in Chapter 2.

It is important to note that this project leans on a previously implemented system for video-surveillance in traffic road environments. The precedent project was also developed in the Multimedia Processing Group of the Signal Theory and Communications department at the Carlos III University.

The main goal of said project was to perform successful anomaly detection in traffic environments. The dataset used for this precedent project is the same as the one that will be used to test the system developed for our project. Moreover, our project will make use of several resources provided by its precedent, as we will later see in this manuscript. In Section 2.4, this anomaly detection project will be further explained.

1.4 Motivation and goals

The main motivation of this project is to improve road safety by providing a technical solution to a problem that currently lacks proper monitoring and regulation.

Traffic accidents affect the whole population, and any improvement in accident rates is good news. Nowadays, there are various technologies that help monitoring and regulating traffic. Some examples are cameras, radar, helicopter patrols, police controls, etc. By adding a new way of monitoring traffic related accidents and events, we will be able to prevent further fatalities in the road environment. In this sense, computer vision offers an easy-to-deploy way to implement such kind of systems. Hence, it is a perfect solution for this problem. However, the use computer vision techniques have been historically limited to video analytics, due to the practical difficulties in their application to real scenarios. Traditionally, taking quantitative measures

from a video flow has been a very tedious, if not impossible, task. This is mainly caused by inevitable challenges such as occlusion due to perspective, lighting or weather conditions.

If we could be able to take quantitative and accurate measures by using computer vision techniques, these techniques would exponentially increase their potential to monitor and regulate traffic environments. Although some cases could also be solved by technologies such as radar, the deployment and maintenance of these technologies is expensive. Therefore, cost effectiveness can be added to the list of motivation points.

Having stated the main motivation points of the project, it can be said that the main goal is to efficiently develop a system that is able to properly detect the desired driving behaviour, in order to prevent potential accidents in the future. With this in mind, we subsequently present three primary objectives:

- The system is able to automatically generate, when the necessary conditions apply, the geometric models used to effectively define the road.
- Using these models, the system is able to perform geometric transformations that allow us to take quantitative measures on a normalised (top-view) environment.
- From these measurements, the system is able to detect when tailgating behaviours are taking place on the scene.

These objectives are crucial, as the main historical problem with computer vision techniques has been the impossibility to take quantitative measures due to limitations on perspective, lighting or other unexpected artifacts.

Next, a list of secondary goals is provided. These goals, while still regarded as important goals, are classified as secondary, as they are less vital than the first group.

- The project implementation is simple and generic, in order for future lines of work to be able to adapt further investigations to the current project.
- The system provides a robust method for estimating homographies.
- The visualization of the results obtained is straightforward and easy to interpret.

1.5 Work Methodology

This project was developed following a very clear methodology, which allowed us to properly implement the different stages of the solution. In Fig. 3 the methodology is displayed as a flowchart, and we will briefly describe each stage in this section.

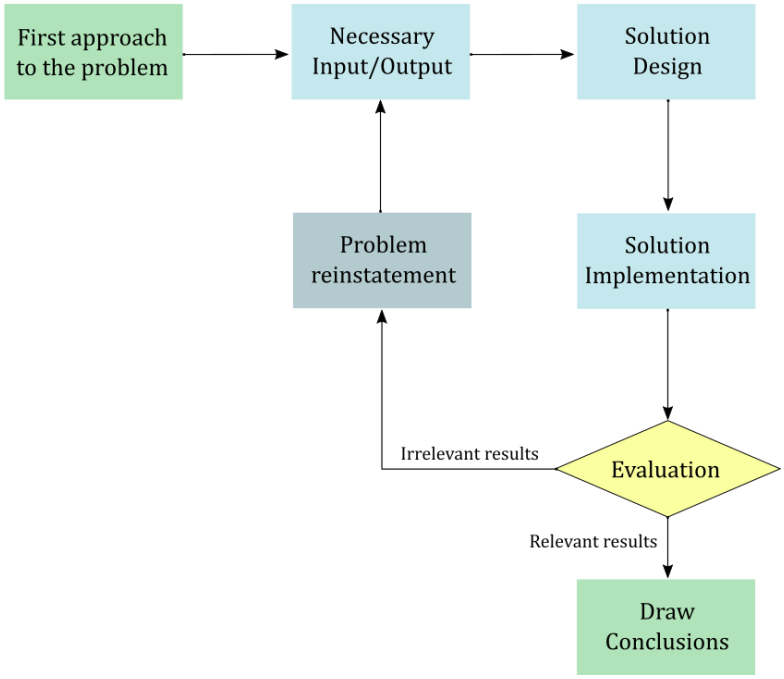


Figure 3. Work methodology flowchart.

The first task to be performed is an initial analysis and understanding of the problem. Once the problem has been clearly identified, we begin to approach the possible solution. For this, we decide what should be the output of our system and what do we need to achieve this output. We gather the resources we have, which can be used as input, and we begin to design the solution. After the solution design is completed, we develop and implement it. Then, an evaluation is performed. During this evaluation, we decide if the results achieved with the implemented solution are relevant. If so, we draw conclusions and we mark the solution as finished. However, if the results are inconclusive or irrelevant, we need to reconsider the problem, and begin the cycle again in order to obtain a relevant solution. Note that this is a generic scheme, and can be therefore used for the overall system, but also applies to every small stage that the project contains. Hence, we will follow this methodology with each individual task during the development of the system.

In some occasions, the available literature will provide us with methods that we can directly use in our implementation. However, the characteristics of the design will require novel contributions in order to fulfil the objectives of the project.

1.6 Novel contributions

The majority of the design for this project was fully implemented for the occasion. However, there are some contributions that are more significant and pioneer. This section contains a brief list of the most important novel contributions provided by the author for this project.

- **Road line detection and classification algorithm:** An algorithm has been fully developed for this system that is able to, given a simple traffic scene background image, detect and classify the white road marks into two different categories (discontinuous lines and border lines). The importance of this algorithm is that it requires very little input and provides a very powerful resource that can be used for further processing. This algorithm is explained thoroughly in Sections 3.3.1 and 3.3.2.
- **Robust homography estimation algorithm:** This algorithm is used to select a proper set of points to calculate a homography. It is based in the RANSAC method (see Section 2.3.2), but it has been develop to fulfil the available resources and optimize computation cost. The developed algorithm takes as an input a set of points (with length $4 < l < 6$) and selects the combination of those points that provides the most robust homography by stating an error based estimation of inliers and outliers. It is explained with more depth in Section 3.3.3.
- **Event specification module:** This is the module of the project devoted to detect tailgating behaviour. Even though it was also fully implemented for this project, perhaps the most important contribution is the theoretical design of the module, as it could be used for the detection of other driving behaviours. As such, it can be considered as a general algorithm to detect and classify driving behaviours. This algorithm is further explained in Section 3.4.5.

The results of the experiments performed for the before mentioned contribution algorithms have been published in an article at the Proceedings of the Advance and Applications of Data Science and Engineer Workshop, an international workshop organized by Real Academia de Ingeniería in Madrid (June 2016) [1].

1.7 Document organization

This manuscript is divided in several chapters, each containing a set of sections development the contents of the chapter. The organization is as follows:

- **Chapter 1:** Introduction. This chapter introduces the necessary bases for the better understanding of the project. It presents the context and environment of the investigation, the motivation and goals that led the author to its development and the work methodology that will be followed to develop the project. Additionally, the novel contributions provided by the author are discussed.
- **Chapter 2:** State of the art and related technologies. In order to facilitate a better understanding of the contents of this manuscript, this chapter provides the reader with an analysis regarding several investigations related to video analytics in the context of traffic surveillance and management. Furthermore, it gives an insight on the technologies involved in the development of this project.
- **Chapter 3:** Technical solution. This chapter presents the technical approach to the presented problem. For this, it first defines the problem and identifies the requirements

and restrictions that need to apply to the system. It also provides the reader with the regulatory context of this area of research, and determines its impact on the presented solution. Then, it thoroughly describes the system, detailing the information about the design and implementation of the different stages of the processing pipeline. Finally, it includes a discussion regarding possible design alternatives.

- **Chapter 4:** Experiments and results. This chapter contains information about the conducted experiments. First, it presents the relevant aspects of the dataset that was used for the experiments, and the experimental setup used to evaluate the performance of the system. Finally, it presents the set of results obtained from these experiments, along with their interpretation.
- **Chapter 5:** Planning and budget. In this chapter, the main details regarding the planning of the project, such as the schedule and the development tools, are presented. After that, the budget information is broken down.
- **Chapter 6:** Conclusions and future work. This final chapter gathers a collection of the conclusions drawn from this project. The manuscript ends with a set of propositions for future lines of investigation.

2 State of the art and related technologies

In the previous chapter, we defined the essence of the problem at hand and present the context and the means with which we will attempt to solve it. However, this research is not isolated. At the time in which investigation started, several investigations had been carried out in the same area of research. The following chapter presents a review on the research performed within the area that falls to this project. We will explain the problems that these investigations found worthy of research, and the different attempts of solving it. To this end, a state of the art analysis will be carried out and documented.

When establishing the area in which this analysis should be performed, it became clear that the focus should be on computer vision techniques. More specifically, the state of the art analysis of this project is focused on video analytics for traffic management and safety.

2.1 Introduction to computer vision techniques

For many years, computer vision has been a key instrument in many variable scenarios. In this section, we provide a brief insight on the concept of computer vision.

2.1.1 Definition and historical context

The human visual system is a rather powerful instrument. Only by looking at a scene, we can evaluate and distinguish with little or no effort the main characteristics that define it. However, despite countless efforts, we are still unable to completely understand how it works as a whole [5]. This is precisely the main goal of computer vision. We can define as computer vision techniques those that try to recover the characteristics that define a three-dimensional environment from an image or a set of images and, with this information, try to model a copy of the studied environment, or at least the relevant information that is desired to perform a certain process [7]. This idea is displayed in Fig. 4.

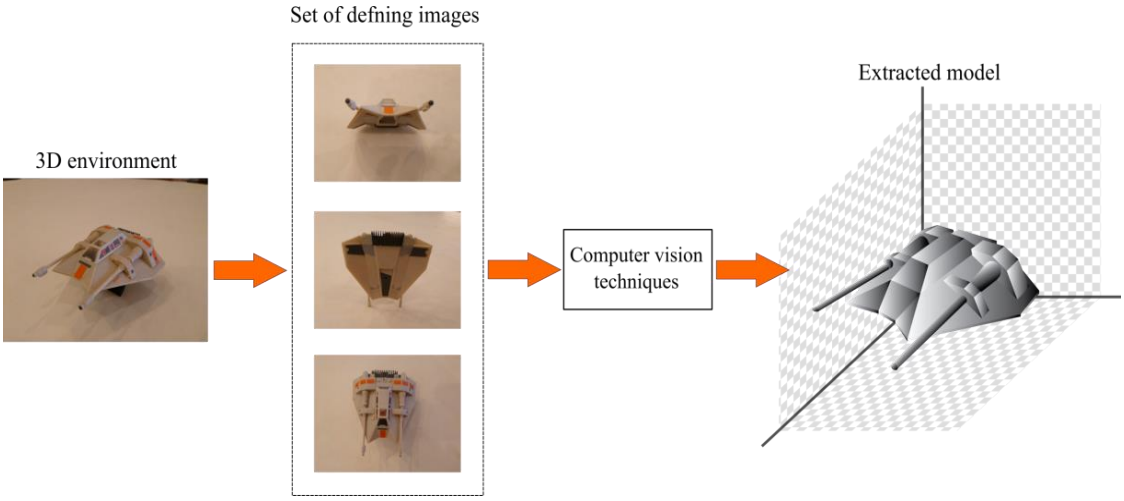


Figure 4. Diagram depicting environment modelling through computer vision.

The history of this kind of techniques is not very extensive. Even though certain experiments that can be categorized as image processing experiments date back to 1801, the vast majority of

the techniques in this area have been developed in the past few decades [8]. Subsequently, this introduction presents the reader with an overview on the recent history of this field [7]:

- **1970s:** During this decade, computer vision was strongly linked to artificial intelligence and robotics, and the main goal was to recover the structure of the world from images, in order to achieve a better understanding of the full scene.
- **1980s:** Diverse lines of investigation led researches to more sophisticated mathematical techniques for performing quantitative image analysis.
- **1990s:** Significant efforts were put into solving the structure from motion problems. Also, notable improvements were made regarding optical flow methods and multi-view stereo algorithms, used to produce 3D surfaces. An improvement also worth mentioning is tracking algorithms. Many investigations regarding active contours, such as snakes, particle filters and level sets were performed during this decade [9].
- **2000s:** In more recent times, the interplay between the fields of graphics and vision has been deepened. In particular, image-based rendering techniques (rebranded by some as computational photography) have played a notable part in the last few years. Another interesting aspect of this decade is the emergence of feature-based techniques for object recognition.

During the last few years, some of the fields mentioned before were improved even more, but one of the most relevant fields to mention here is machine learning, which has been recently used to solve computer vision problems. Recently, fields like deep machine learning and Convolutional Neural Networks (CNNs) have been proved to improve the state-of-the-art regarding most computer vision related tasks [10]. Nowadays, there is a large amount of applications of computer vision in several areas.

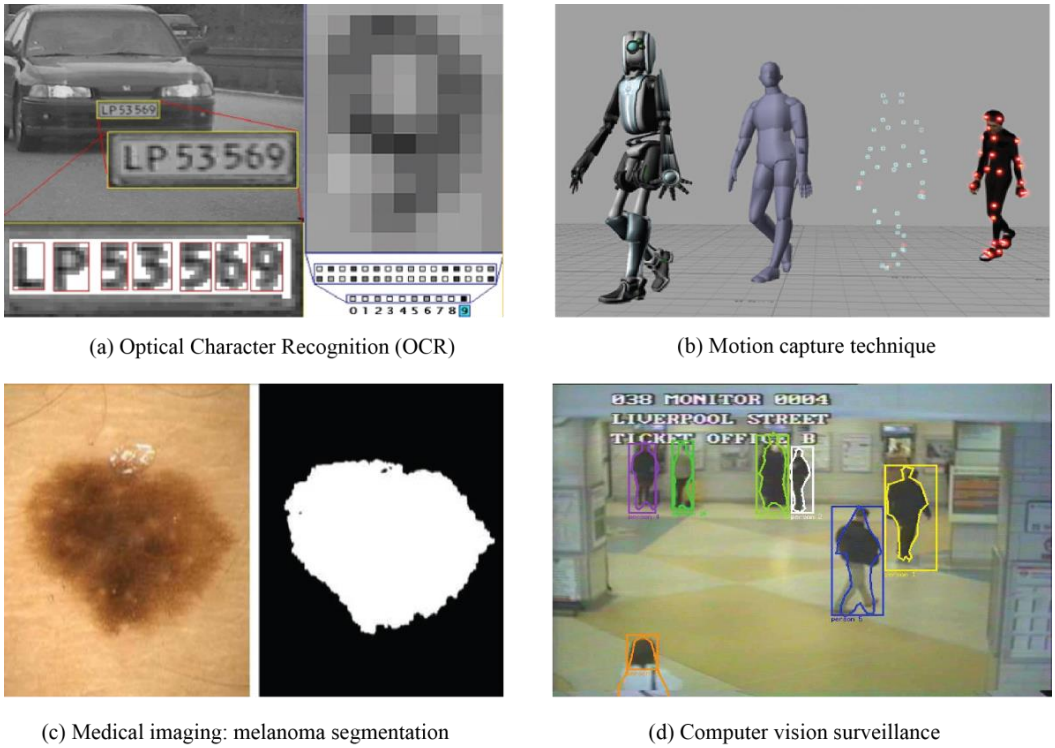
2.1.2 Computer vision applications

There is a common misbelief regarding computer vision techniques, usually from people that have never worked in the field, that computer techniques are effortless. This may be due to the early research performed on artificial intelligence areas, as it was thought that the cognitive parts of intelligence, those that refer to logic and understanding, were inherently more difficult to imitate than the perceptual parts [11]. However, not only is computer vision a complex and complete area of research, but also useful and applicable to many different scenarios.

The range of computer vision techniques is wide and assorted. Subsequently, we present a list the most relevant applications of said techniques:

- **Optical character recognition (OCR):** Technique consisting on recognising certain characters (Fig. 5.a shows license plate identification using OCR). It is used for reading postal codes or license plates, for instance. Optical character recognition is especially useful when the information needs to be interpretable both by humans and machines [12].
- **Motion capture:** This technique is widely used in current movies. The use of retro-reflective markers allows cameras to capture precise movement of actors. This information is later used to recover full articulated body motion [13]. Fig. 5.b shows the transition from the retro-reflective markers (right) to a computerized model that replicates human motion (left).

- **Medical imaging:** Medical images can be machine analysed to diagnose certain diseases. One example in this line of research is the use of segmentation methods for melanoma diagnosis in dermoscopy images [14], like it can be seen in Fig. 5.c.
- **Surveillance and safety:** Monitoring an environment to detect possible threats to individuals is also performed via computer vision. Intruder detection systems and highway traffic monitoring are examples of this. Fig. 5.d displays an environment where different individuals are tracked to detect unusual behaviour.



(a) Optical Character Recognition (OCR)

(b) Motion capture technique

(c) Medical imaging: melanoma segmentation

(d) Computer vision surveillance

Figure 5. Some computer vision applications. (a, c & d) taken from [15][16][17], respectively.

There are many more uses to this area, like fingerprint recognition, visual authentication or face recognition [7], but the main interest for this project is surveillance. Computer vision surveillance techniques will allow us to develop a system that effectively detects tailgating behaviour in a road environment. However, these techniques also have some limitations that need to be taken into account when developing such system.

2.1.3 Computer vision limitations

Computer vision techniques are incredibly useful for defining the main characteristics of a set of images. However, there are some limitations that strongly difficult this type of tasks. First, any analysis that we may perform in an image is altered by the image's resolution or any noise that it may have for whatever the cause. Such restricting factors can be mitigated, for example, by using better quality equipment. Regrettably, there are some other limitations to these techniques that are even harder to overcome. In this section, we briefly describe those of them that have affected the development of this project:

- **Point of view:** The point of view in which the traffic cameras are located is a priori unknown and depends on the particular scene and road location. Therefore, as we do

not know the internal and external parameters of the cameras, we cannot establish a general transformation that allows us to interpret the information in the image.

- **Occlusion:** One of the main disadvantages of building a 3D model with information of just one point of view is that if an object is located, partly or fully, behind another object in the image, it will be impossible to retrieve. Therefore, visual information hidden by occlusions is inevitably lost.
- **Scale:** This is also a challenge for this project, as we need to take measures from the scene, but we do not know the real sizes of the objects captured by the camera. As a result, we cannot determine the real parameters of the scene unless we identify an element for which we know its real dimensions.
- **Variable background:** The background in traffic scenes is very unstable. Not only does the change of light affect the scene, but also other external factors do. For instance, certain vehicles are intermittently becoming part of the background and the foreground. These changes make foreground detection more challenging.

For the correct development of the project, these limitations need to be accounted for and diverse techniques will be used to avoid, in the most effective way possible, the artifacts, errors or misleading results that they may produce.

2.2 Previous works in road traffic surveillance

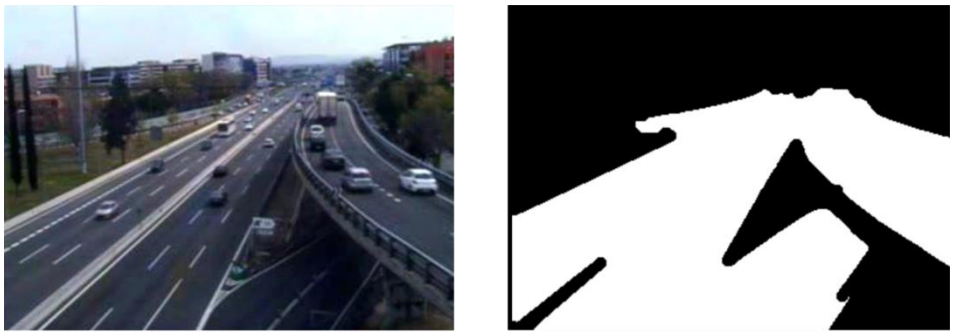
This section of Chapter 2 is centred in providing an insight of the research works that had been previously carried out in the field of road traffic surveillance.

2.2.1 Relevant techniques for traffic management and safety

In the following section, we will firstly focus on anomaly detection techniques, as this area of research represents the foundations of our project. Video analytics and anomaly detection techniques have been successfully implemented to properly detect some events, like traffic congestion, accidents or other anomalies that may occur in road scenarios. We will discuss the different techniques regarding anomaly detection shortly. In addition, this chapter also presents the field of road segmentation, which has been crucial in the extraction of the road models used in this project. After that, we will also explain a method used for the estimations of homographies, as well as another method to add robustness to these homographies, which were vital to this project.

2.2.1.1 *Road segmentation*

The segmentation of a road in a traffic scene is a rather useful task in many computer vision applications for traffic management. For instance, it can be used for security systems that monitor the road from within the vehicle [30]. However, the topic that is of the most interest to this research is general traffic surveillance. There are multiple applications of road segmentation in traffic surveillance, like advanced traffic assistance, lane departure and aerial incident detection [31].



(a) Traffic scene example (b) Road segmentation example

Figure 6. A traffic scene and its segmentation represented with a binary image.

A proper segmentation of the road allows any system to wrap the regions in which the analysis has to be performed, as traffic events usually occur within the limits of the road. This is very interesting, as it speeds up the analysis of the videos. There are several lines of research that study road segmentation for this purpose.

The research presented in [31] performs road segmentation relying on super-pixel (e.g. a small connected region of pixels that are homogeneous with respect to some criteria) detection. In each super-pixel, features are extracted and fed to a support vector machine (SVM) classifier. Then, this classifier decides whether the area within the analysed super-pixel does or does not represent a part of the road. The process displayed in Fig. 7 represents the two modules of this method. The first is oriented to detect the super-pixels by a method of edge density estimation. The second is destined to extract the features (colour, texture homogeneity, motion estimation) and feed them to the SVM to define the road.

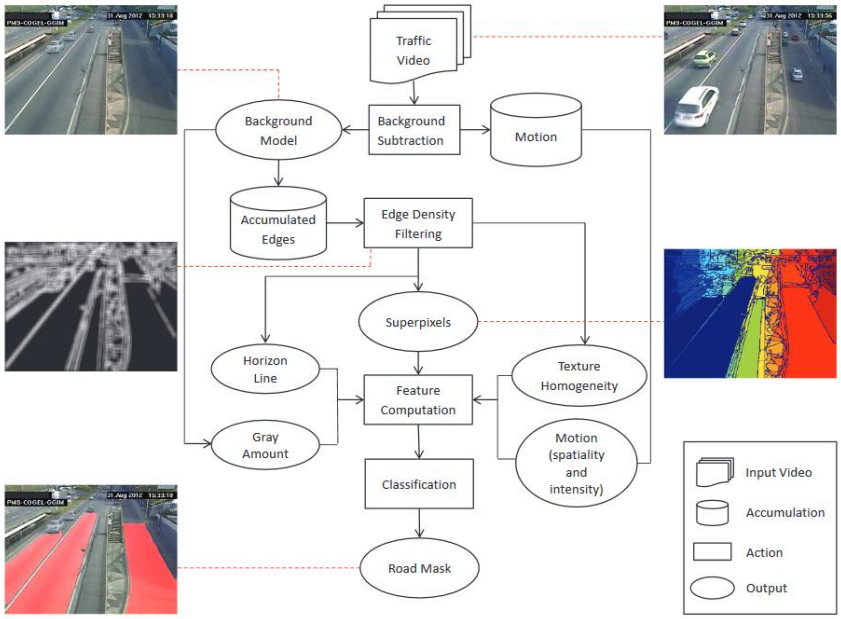


Figure 7. Outline of the road segmentation system [31].

Besides this investigation, there are others that use different techniques to address a similar problem. One example is the use of a low-level object tracking system to produce vehicle motion trajectories. This trajectories allow us to define the road and its parts, for instance, we can detect lane centres and classify lane types. Additionally, this method can be used to detect anomalous trajectories. For the interested reader, the complete method and further information about its functioning can be checked in [32].

Even though this latter method is not the very same as the one we mentioned in Section 1.3 (and will be later explained in Section 2.4), there are some similitudes, as the segmentation is determined based on object trajectories of moving vehicles along time. Therefore, this type of method is more relevant to our project.

2.2.1.2 *Anomaly detection techniques*

In traffic safety, monitoring incidents is a quite important task. Proper monitoring leads to rapid response to accidents and helps to improve the diagnosis. As a result, a system that is able to detect anomalous events is very valuable for traffic safety [18]. In this particular field, computer vision has a lot to offer, as it provides a quick way to recognise these problems via anomaly detection. Anomaly detection systems in traffic management use several methods to achieve a common goal: to detect any behaviour, out of the usual, that may affect to road safety. In this section, we present some of the techniques found in the literature.

2.2.1.2.1 **Clustering and classification of object trajectories**

Research regarding surveillance systems has focused on object trajectory representation schemes, usually from a high-level perspective [19]. These systems are usually capable of producing information about the motion of the objects in the analysed scene [20]. They usually rely on other low-level modules for obtaining object-based trajectories and organizing them in tracking schemes [21].

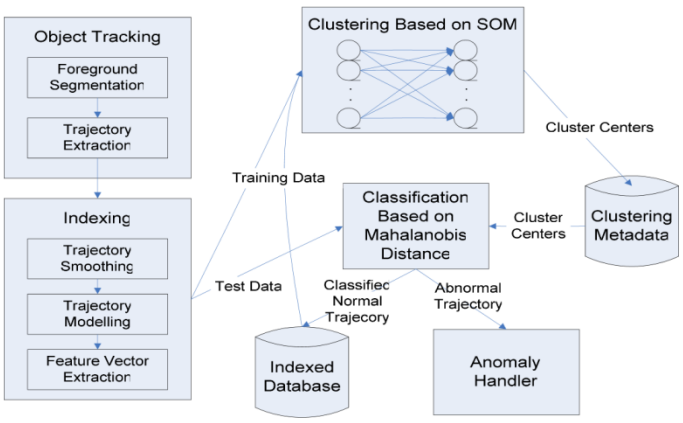
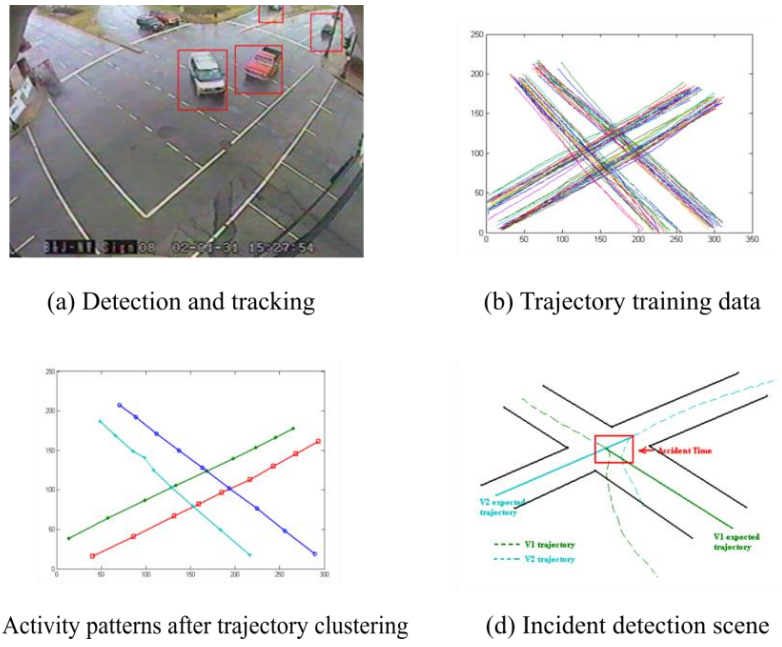


Figure 8. Overview of system architecture for learning object trajectories [19].

The basic idea is to identify the trajectories of the different objects that are present on a certain scene by discovering the clusters in the data available. From this, we gather the most habitual trajectories. Then, during test, we compare the newly found trajectories to the clusters obtained during training, and classify them as anomalous depending on the similarities between them.

The research performed in [19] proposed techniques for clustering and classification of object trajectory-based videos using spatiotemporal function approximations. The goal was to learn motion patterns by reducing the number of dimensions of the data that the system had to handle. In other words, the main objective was to represent the high-dimensional data space with coefficient feature space, hence obtaining a low-dimensional data space, much easier to handle. An overview of the system can be observed in Fig. 8. This technique has shown improvements in the accuracy of the results compared to experiments learning from the high-dimensional data space.

The focus in [22] is on vehicle motion analysis. The activity of the scene is defined by the use of the before mentioned clustering techniques. First, each vehicle's trajectory and its motion features are extracted from the scene, establishing a tracker for each vehicle (Fig. 9.a). With this data, the general information about the trajectories is extracted (Fig. 9.b), and then used to determine the activity patterns that define the scene (Fig. 9.c). Finally this data is used to define the events, and normal and abnormal events are selected using a threshold. This is displayed in Fig. 9.d, where the expected usual behaviour of the vehicles involved in the event (V1 and V2) is represented with a straight line, whereas the actual trajectories detected are shown in dotted lines. In this case, the system evaluates the trajectories and classifies the event as unusual. One can then use multiphase linear regression techniques to classify the different unusual events that have been detected.



(a) Detection and tracking (b) Trajectory training data

(c) Activity patterns after trajectory clustering (d) Incident detection scene

Figure 9. Stages of event detection using vehicle motion analysis in [22].

2.2.1.2.2 Microscopic traffic variables

There is another technique addressing the problem of anomaly detection that is worth mentioning here. This technique takes into account changes in the variability of microscopic traffic variables [23]. Such variables may include relative speed, inter-vehicle time gap and lane changing.

In order to detect traffic anomalies, this technique first focuses on detecting transient anomalies. These occur at the beginning of the variations in traffic patterns. Detection of transient anomalies is a key to detect anomalies, as they could be the sign of a major incident about to occur. There is plenty of literature regarding incident precursors, which are the anomalies that lead to actual incidents. Such investigations use microscopic traffic variables obtained from road-side infrastructure [24], like inductive-loop traffic detectors. However, the performance of these algorithms highly depend on the position of the loop detectors, as anomalies happening too far away from the detector may be exceedingly delayed or even not detected at all.

Nevertheless, the algorithm proposed in [23] makes use of the advances in vehicle-to-vehicle and vehicle-to-infrastructure wireless communications [25] to effectively improve real-time measuring of microscopic variables. This allows the algorithm to achieve 100% detection rates and negligible false alarm rates with partial microscopic traffic information from as few as 20% of vehicle population. Hence, there is no longer the need for locating detectors at locations where anomalies tend to occur.

2.2.1.2.3 **Machine learning improvements**

The use of machine learning techniques allows us to develop non-parametric, change-adaptable algorithms, which are also portable across applications [26]. Machine learning is a discipline destined to the implementation of algorithms (mainly induction algorithms) that are said to *learn* [27]. The term *learning* does not have the same connotation as usual cognitive learning. Machine learning algorithms make use of labelled data to train a system to perform a task (*learning*). This system can be later used to predict or detect certain behaviours based on unlabelled data.

One of the most known problems in computer vision is the one that tries to recreate a model of the human visual system. There probably exist several hundred models of the visual cortex, but machine learning algorithms allow us to obtain more comprehensive models than other techniques [28]. Experiments performed in [26] have yielded promising results regarding traffic anomaly detection using machine learning techniques. When applying their algorithm to traffic image sequences from their dataset, they were able to outperform previous anomaly detection algorithms. The experiments performed have proven that the proposed algorithm, *Kernel-based Online Anomaly Detection* (KOAD), outperforms previous algorithms like *One-Class Neighbour Machine* [29] (OCNM).

2.2.1.3 ***Quantitative measurement techniques***

Traditionally, computer vision has been severely limited regarding quantitative measurements. However, there have been some attempts to tackle this matter.

The research performed in [33] proposes a method to measure vehicle's speed by using just a digital camera and a computer. The analysis they perform on the video allows them to identify the vehicles by their license plates. Once the vehicles are identified, they estimate the distance covered by the vehicles between frames and use geometrical information of the camera-road system (see Fig. 10) to estimate the speed at which they are circulating.

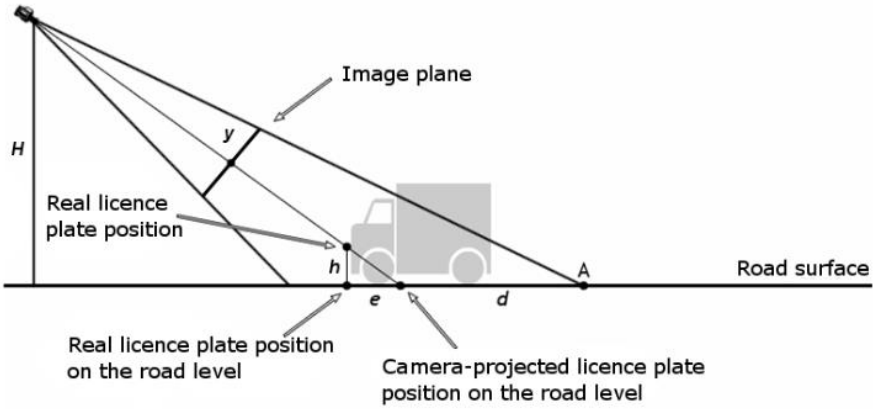


Figure 10. The principle of vehicle position measurement with a single camera. The distance *d* of a vehicle from the reference point *A* in the camera's field of view is proportional to the position of the vehicle in the image plane *y*. Figure taken from [33].

This research also points out the advantages of computer vision based systems over radar systems, as the latter are considerably more expensive and cannot be widely used. The method proposed in [33] is an important inspiration for the development of this project, as it gives an idea on how to calculate distances based solely on a video flow.

Another investigation implemented a method to regulate traffic by detecting speed violations and issuing citations accompanied by supporting video evidence [34]. Some of the main advantages that the research highlights are that it is less costly than current radar systems and that it facilitates monitoring of traffic in multiple-lane roads. Another gain of this type of system is, as they state, that as cameras are passive technology this monitoring is undetectable by current in-car devices. Therefore, it does not allow the driver to prevent sanctioning in monitored areas while violating regulations outside these areas.

2.3 Technologies involved in this project

Besides the aforementioned research works, there are several other techniques that influenced the development of this project. In this section, we present a summary of the algorithms and methods that were used, partially or fully, at some point in the development of the research carried out.

It is worth mentioning that these technologies were, in most of the cases, initially proposed for different purposes than the ones we have pursued here. The adaptation of these methods, if relevant, will be later explained in Chapter 3.

2.3.1 Estimation of Geometric Transformations between images

There are many systems that rely on detection and estimation of geometric transforms between images for very diverse purposes. For instance, a transformation constraint can be used to resolve occlusions in crowded scenes and accurately track people moving in the analysed scene [35]. Additionally, one of the methods for quantitative measurements mentioned in Section 2.2.1.3 also makes use of transformation matrices to improve their performance [34].

In this project, we are particularly interested on computing projective transformations, also known as homographies. A homography is a transformation defined between two planar spaces. In the context of this investigation, it is defined with a simple 3x3 transformation matrix (*H*). In

order to better understand the usage of homographic transformations in this project, we subsequently present an explanation of the method to obtain such transformation matrix.

First, it is important to clarify that in order to determine the matrix that maps one set of points belonging to a certain plane to another set of points belonging to a different plane, we need to have full knowledge of both plane's coordinates.

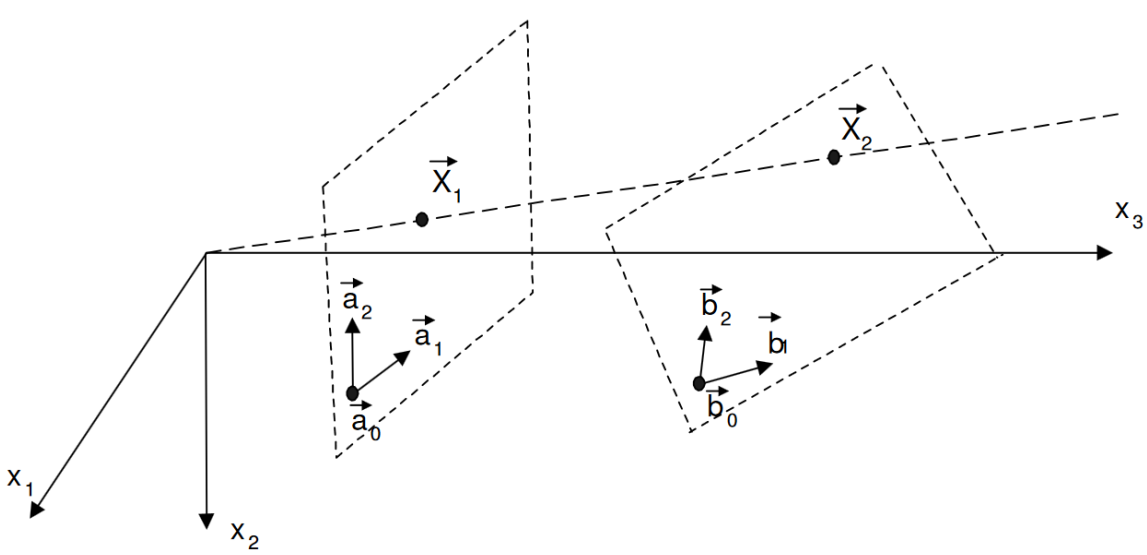


Figure 11. 3D coordinate frame with two arbitrary planes [36].

If we consider a 3D coordinate frame like the one depicted in Fig. 11 we can define two arbitrary planes in said space. For this, we choose one point ($\overrightarrow{a_0}$ *and* $\overrightarrow{b_0}$) and two basis vectors (($\overrightarrow{a_1}$, $\overrightarrow{a_2}$) *and* ($\overrightarrow{b_1}$, $\overrightarrow{b_2}$)) for each plane.

Then, a straight line from the origin intersects with both planes at points $\overrightarrow{X_1}$ and $\overrightarrow{X_2}$, respectively. We can henceforth define said points as:

$$\overrightarrow{X_1} = p_1\overrightarrow{a_1} + p_2\overrightarrow{a_2} + \overrightarrow{a_0} \quad (1)$$

$$\overrightarrow{X_2} = q_1\overrightarrow{b_1} + q_2\overrightarrow{b_2} + \overrightarrow{b_0} \quad (2)$$

From (1), we can simplify and express:

$$\overrightarrow{X_1} = (\overrightarrow{a_1}, \overrightarrow{a_2}, \overrightarrow{a_0}) \begin{pmatrix} p_1 \\ p_2 \\ 1 \end{pmatrix} = A\vec{p} \quad (3)$$

On the other hand, the relation of the points from the origin can be expressed as:

$$\overrightarrow{X_1} = \alpha(\vec{q})\,\overrightarrow{X_2} \quad (4)$$

where $\alpha(\vec{q})$ is a scalar that depends on the second plane. From (3) and (4), it yields:

$$\vec{p} = \alpha(\vec{q})\,A^{-1}B\vec{q} \quad (5)$$

Note that vectors $\vec{p}$ and $\vec{q}$ both have 1 as the third coordinate, so the function of the scalar $\alpha(\vec{q})$ is mainly to force $\vec{p}$ to have a unit third coordinate. Therefore, if we move to homogeneous coordinates, we can avoid this scalar and substitute it by any arbitrary scalar c.

Therefore, we can rewrite (5) as:

$$\overrightarrow{p^h} = c\, A^{-1} B \overrightarrow{q^h} \tag{6}$$

where $\overrightarrow{p^h}$ and $\overrightarrow{q^h}$ are homogeneous 3D vectors.

As a result, we obtain the term that maps one set of homogeneous coordinates to another set of homogeneous coordinates, each expressed as a function of the basis that we chose to define the planes in which they are located. This term is named H, and it represents the homography matrix.

Summarizing:

$$\overrightarrow{p^h} = H \overrightarrow{q^h} \tag{7}$$

where H is the homography matrix that maps the points in the different planes, and can be expressed as follows:

$$H = c\, A^{-1} B \tag{8}$$

Note that we can retrieve the original vector $\vec{p}$ any time from the homogeneous coordinate vector $\overrightarrow{p^h}$.

There are many applications for homographic transformation in computer vision. One example is to build a complete image of a certain object or environment from various small images. This requires image to image mappings, and therefore involves homographies between pairs of input image to produce the resulting compound image as an output [36]. This technique is displayed in Fig. 12.

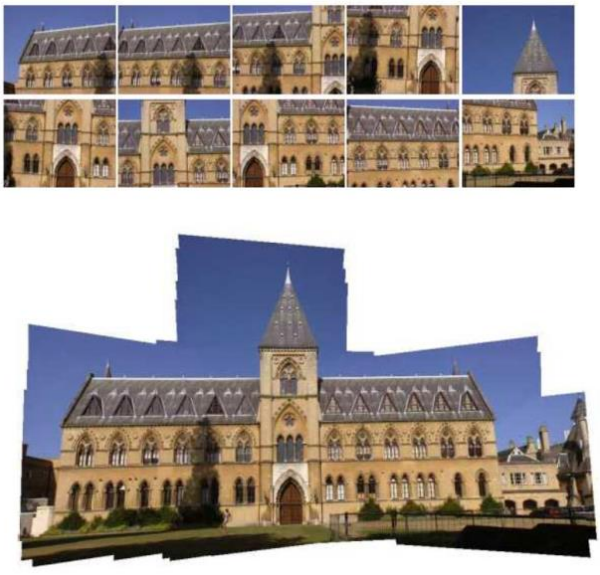


Figure 12. Mosaic technique example [36].

Other methods like removing perspective distortion in computer vision, or rendering textures and computing planar shadows in computer graphics, also make use of homographic transformations.

Once the concept has been clarified, another question arises regarding homographies, and this is how we obtain such homography matrix in real systems. For this project, the *Direct Linear Transformation (DLT)* is used. This method applies Singular Value Decomposition (SVD) to obtain the resulting matrix H. The interested reader is referred to [37] for a detailed description of this method.

2.3.2 Robust Model Estimation through RANdom SAmple Consensus (RANSAC)

Another relevant technology that is worth discussing is the robustness method described in [38]. The RANdom SAmple Consensus (RANSAC) method provides a way of fitting a model to experimental data. This method is able to interpret data even when it has a notable number of errors or outliers. As a result, it adds robustness to the dataset that we are using, discarding the samples that are less likely to be a part of the ideal set, marking them as outliers (Fig. 13).

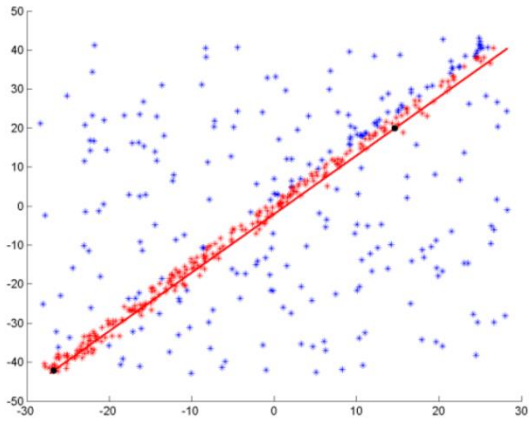


Figure 13. Example of the result of the RANSAC algorithm, where the inliers are marked in blue and the outliers in red [39].

The RANSAC algorithm can be summarized in the following steps:

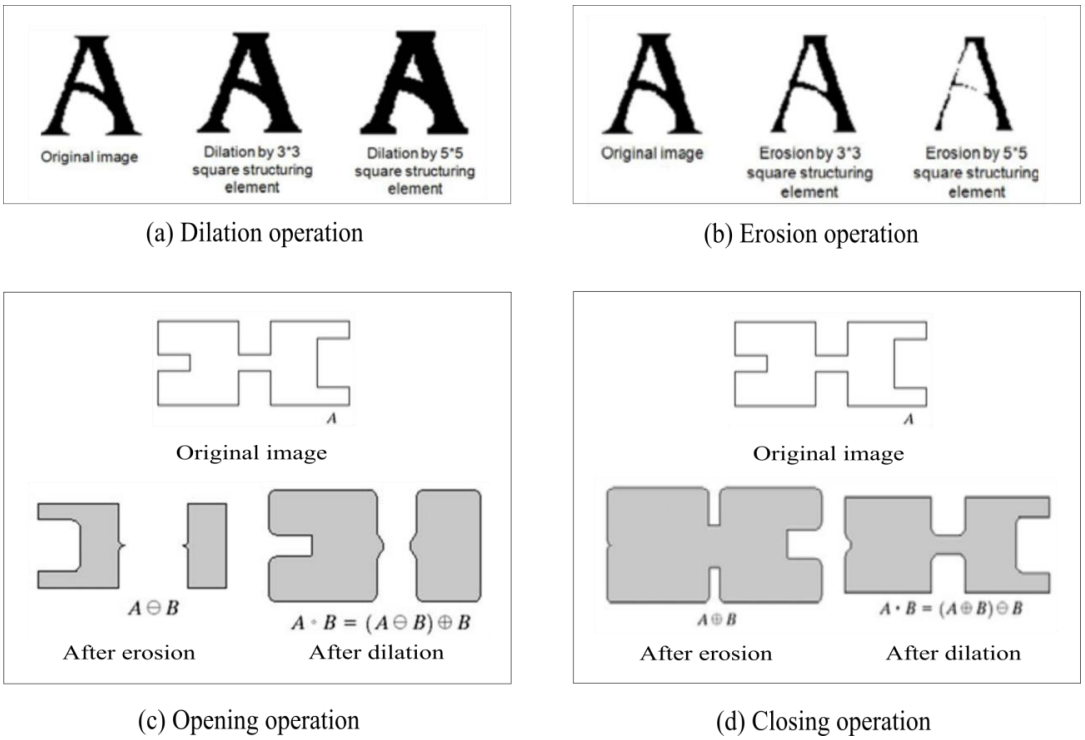
1. Given the original set of data, a subset is randomly selected, forming the hypothetical inliers. The set has to be sufficiently large to determine the model, but is selected as the minimal set that achieves that.
2. A fitting model is then computed using only the set of hypothetical inliers.
3. The data that was not part of the hypothetical inliers is then tested using the model obtained in step 2.
4. A loss function is specifically establish for the model, and those points outside the hypothetical inliers that satisfy an error threshold imposed by the loss function are considered as part of the consensus set.
5. The model is classified regarding its quality depending on the number of points classified as part of this consensus set. If it surpasses a certain threshold it is saved, otherwise it is discarded.

This algorithm is iterative. It is repeated a certain number of times to find the model with the highest quantity of points in the consensus set. It is very useful for automatic image analysis, as this is an area where the data is usually provided by detectors that tend to make errors systematically. One example of method that uses this method is the research discussed in 2.2.1.1, centred on road segmentation [32].

In this project, a technology based in RANSAC is used to perform robust identification of key points. We need an accurate detection of these key points, as they will be used to estimate a transformation between the original environment and a new one where measurements are feasible.

2.3.3 Morphological image processing

To put an end to this section and to this chapter, we will discuss certain techniques that facilitate image processing in some key cases. These are the morphological processing techniques, which deal with the shape (or morphology) of objects or features in an image. As this area is a broad and well-researched one, we will only briefly comment on a few aspects of morphological processing. For a deeper insight on the topic, the reader is referred to [40].



(a) Dilation operation (b) Erosion operation

(c) Opening operation (d) Closing operation

Figure 14. Morphological image processing: basic operations [40].

As stated, this section will be devoted to briefly explain a few basic morphological operations that will be relevant for the development of this project.

- **Dilation:** The dilation operation (Fig. 14.a) dilates a binary image based on a structuring element. If the structuring element is bigger, the dilation experimented by the image will also be greater. Dilation is used for repairing breaks and intrusions.
- **Erosion:** The erosion operation (Fig. 14.b) does the opposite of the dilation operation. The extent to which the image is eroded is also related to the size of the structuring element used. Erosion can be used to split joint objects or to strip away extrusions.
- **Opening:** The opening operation (Fig 14.c) combines erosion with dilation. First, it erodes the original image, only to later dilate it. This operation is used to smoothen the contour of the objects in an image, and also eliminates thin parts of the image.

- **Closing:** The closing operation (Fig. 14.d) combines dilation with erosion. It performs the same operations as the opening, but in reverse order. This time, dilation is performed first, thickening any thin parts and closing small holes. Then, an erosion operation is performed.

2.4 Previous system for anomaly detection and video analytics in road traffic surveillance

The system on which this project leans had as a main objective to improve road safety, mobility and traffic management. In general terms, the focus was on the detection of traffic incidents in roads. The system requires certain time to learn the usual behaviour of the vehicles in the scene under analysis. By doing so, it is able to detect unusual behaviour later on, and this can be classified as an incident. The general pipeline of the system is displayed in Fig. 15.

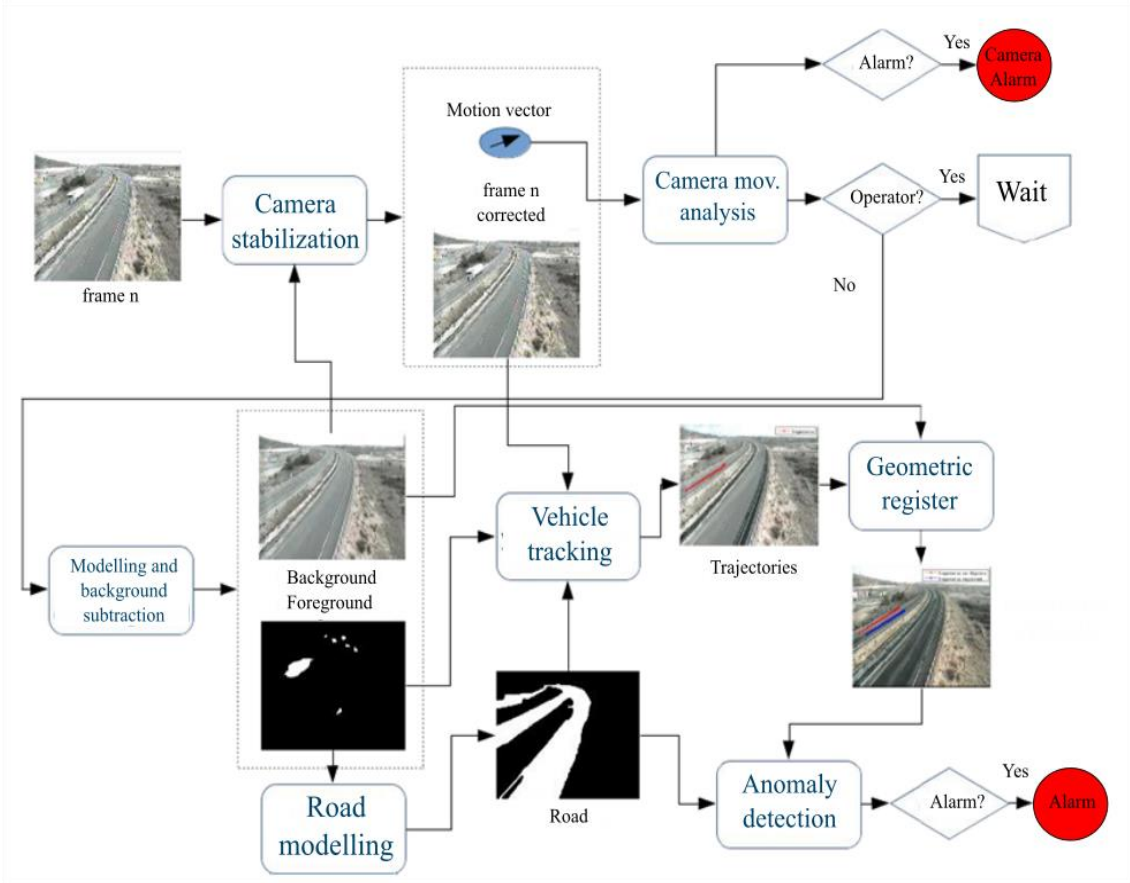


Figure 15. General pipeline of the incident detection system [4].

The system considers a **background model**, in which an image representing the scene without the vehicles is stored, and a **road model**, in which the system stores a binary mask indicating the areas of the image that are classified as roads. Both models are initialized with the first frame of the video: the background is set to the first frame and the road model begins with an empty mask. After that, the models are updated every frame and they detect potential

anomalies, as it is shown in Fig. 15. Each step of the process will be subsequently explained briefly:

1. The camera motion is stabilized by estimating the displacement between the current frame and the background model.
2. The camera motion is then analysed in order to detect one out these three possible situations:
    a. The camera motion is exaggeratedly high and doesn't follow any usual pattern followed by the operator: e.g. strong winds or an accident that hits the post in which the camera is placed. In this case, the system generates an alarm for unusual movement of the camera.
    b. The camera motion is willingly caused by a camera operator. The system then deactivates until the manual operation is completed, and then it starts to generate the background and road models again.
    c. The camera motion is small and can be compensated. In this case, the system corrects the current frame in order to stabilize the video and the process continues.
3. Working on the current frame, and once it has been corrected, the modelling and background subtraction module updates the background model and detects the foreground mask, which contains the regions of the image that are not considered background, this is, the vehicles or other moving elements external to the road.
4. The foreground mask and the road model, along with the corrected frame, allow the module of vehicle tracking to compute the trajectories that apply to the vehicles in the scene.
5. As the camera location usually does not exactly match a previous view, the geometric register module is assigned with the task of aligning the current view with a previous one, and then normalizes the trajectories so that they are comparable from both views.
6. As a final step of the process, the anomaly detection module handles the normalized trajectories, stored in a file of vehicle trajectory trackers, and attempts to detect any potential traffic incidents and, when relevant, generate the necessary alarms.

The interested reader is referred to [4] to obtain more detailed information about this project or any of its modules.

It is worth mentioning that for our project we will use many of the data resources generated by this system. The most relevant data resources for our project are listed below:

- Corrected video flow: Video flow of the scene with the vehicles passing through, corrected for vibrations.
- Corrected background video flow: Video flow of the scene displaying just the background, corrected for vibrations.
- Road mask models: Binary models defining the area representing the road for each scene.
- Vehicle trajectory trackers: File containing the trajectories of the vehicles passing through the scene, stored in bounding boxes with four values (horizontal coordinate, vertical coordinate, width and height).

3 Technical solution

The main goal of the project presented in this manuscript is to detect dangerous driving behaviours, namely, tailgating, to prevent potential traffic incidents that may result in a decrease of traffic safety. In order to do this, the system proposed makes use of various techniques to implement a solution for tailgating detection. For that end, the first step in our proposal aims to find geometric models that are able to define the relevant elements of the scene, so that further processing can be made on said scene to find the desired behaviours.

This chapter is devoted to present the technical solution to the presented problem. The different modules that the project includes are explained and a detailed description of the implementation is provided. Finally, an analysis concerning possible alternative designs is presented.

3.1 The problem of tailgating

Reckless driving represents a clear threat to road safety. In particular, tailgating has been regarded as one of the behaviours that concern the average population to a greater extent, only surpassed by drinking and driving, red light running and not wearing seatbelts [41].

Tailgating, defined as a driving behaviour, occurs when a vehicle is driving too closely behind the vehicle in front [42]. However, as simple as definition might make it look, it is not easy to detect. In this manuscript, we propose a system that will be able to detect tailgating behaviours in a traffic video flow.

3.1.1 Problem definition

This section is devoted to briefly clarify the terms of the investigation that will take place in this project. The departing point of this project is a video flow. In this video flow, several traffic scenes can be appreciated. By making use of the technologies previously explained in Chapter 2 and some of the output from the system described in Section 2.4, we will develop a detection system based on vehicle trajectories and geometric models of the road.

Therefore, we need to determine the input and output of the system. After a thorough analysis of the problem, we established a basic structure that can be summarized as follows:

- **Subsystem 1:** Devoted to define the road and its elements.
    - ***Input:*** Output from system defined in 2.4 (background image defining the road scene).
    - ***Output:*** Geometric models of the road.
- **Subsystem 2:** Destined to detect tailgating behaviour.
    - ***Input:*** Geometric models from subsystem 1; output from system defined in 2.4 (corrected video flow, background video flow, vehicle trajectory trackers).
    - ***Output:*** Infraction detection.

3.1.2 Requirements and restrictions

Before we carry on to a deepened description of the implementation and details of the system, we need to establish the conditions under which the system is developed. This section discusses the requirements needed for the presented system, along with a general overview of its restrictions.

3.1.2.1 *Requirements*

This brief subsection contains a list of the requirements that this project assumes:

- The system is provided with certain input from the system described in Section 2.4. Namely:
    - A video of the scene is provided, with the relevant corrections derived from unwilling camera movement.
    - A video flow with the updated background for each frame is provided.
    - A dataset with the trajectory trackers from the vehicles of each scene is provided, with the necessary information for further processing.
- The camera angle in the videos provides a clear and unobstructed view of the road, with a perspective wide enough to contain several vehicles in the same lane.
- The quality of the image has to be sufficient for our method to work out a proper background subtraction based on both the original video flow and the background video flow.
- The lighting influencing the scene at each moment of the video must not severely modify the image.

If any of these requirements is not fulfilled, the system is not guaranteed to function properly.

3.1.2.2 *Restrictions*

Along with the requirements defined before, the project has certain implicit limitations, which are inherent either to the approach to the problem or to the video material with which we will develop the project. The most significant ones are listed below:

- Vehicles occluded by other vehicles or by external objects will not be taken into account for further processing. Therefore, the system will not detect cases of tailgating involving vehicles affected by occlusion.
- Measurements of different features will be affected and therefore limited by the resolution of the image. Certain level of uncertainty is expected when calculating any feature of the system.
- The lack of a labelled database will difficult the task of objectively evaluating the performance of the developed system.
- The influence of the lighting, camera motion, weather or other external incidences in the scene will affect the characteristics of the image, even if it does not completely break the functioning of the system.
- The restrictions and limitations of the resources provided from external sources (see 2.4) will unavoidably propagate to the system.

These restrictions will be addressed in future lines of investigation, and the main goal will be avoiding them to obtain a better functioning system. In Chapter 6, we will discuss how some of these restrictions could be dodged.

3.1.3 Regulatory framework

The framework of this project concerns different aspects of the road environment, and it is therefore subject to certain traffic regulations. In this section, we explain which regulations are applicable to our area of research, and we will comment the relationship with our project.

Although vehicle tailgating represents one of the more frequent causes of traffic accidents [47], most countries do not present a clear regulation on this issue.

A clear example of this issue is Spain. Incidentally, Spain is the country in which this project was developed, and the lack of regulation on the matter affected the decision of initiating this research. Institutions in Spain present recommendations on the distance that is considered to be safe in different situations (Table 1), but no official regulations exist. This is also an encouragement for this project, as despite the fact that we cannot establish a clear threshold on when an infraction is being committed; we have initiated the necessary path for this kind of behaviours to be monitored and regulated.

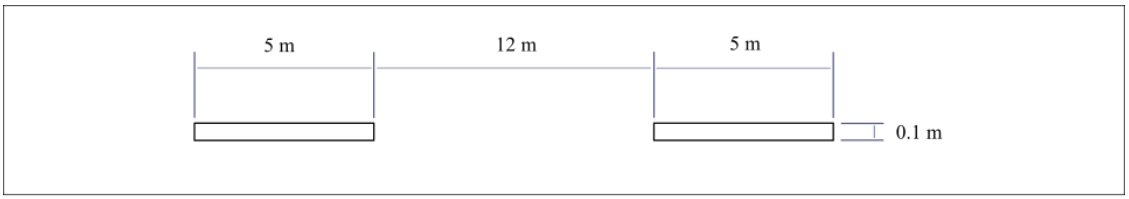
Dry roads		Wet roads	
Speed	**Distance**	**Speed**	**Distance**
50 km/h	25 m	50 km/h	50 m
90 km/h	81 m	90 km/h	162 m
100 km/h	100 m	100 km/h	200 m
120 km/h	144 m	120 km/h	288 m

Table 1. Recommendations for security distance in Spain for dry and wet roads [43].

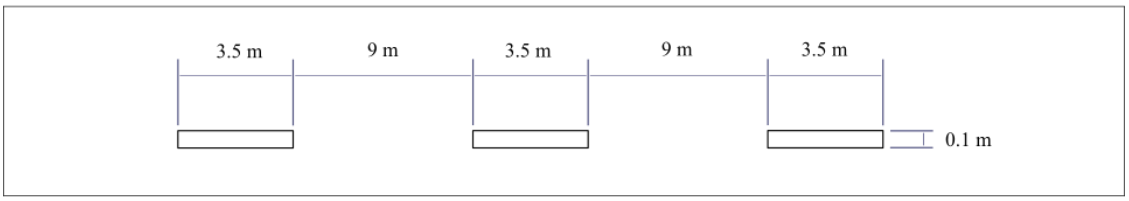
However, there are certain aspects of traffic regulations that are relevant to this project. As it will be later explained in Section 3.3, we need to know some fixed information about the elements of the road in order to extract proper geometric models that are able to define the scene. These elements are the discontinuous lines of the road, for which we need to know the size, and the lanes present in each scene, for which we need to be aware of the width. To conclude this section, we offer a brief summary of the current regulation regarding these elements in Spain.

3.1.3.1 *Discontinuous lines*

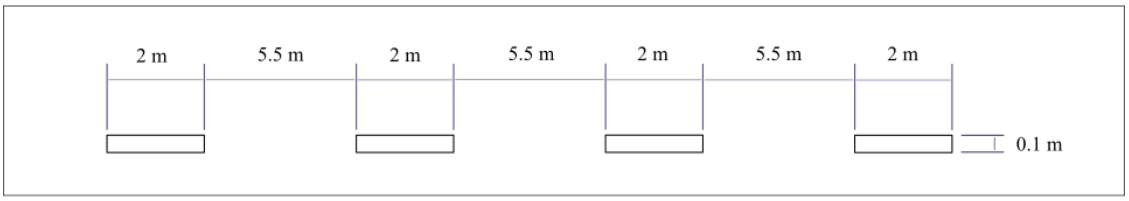
According to the current regulation, the size of the discontinuous lines depends on the speed limitation that the road is subject to. There are three speed intervals at which the regulations change. This is displayed in Fig. 16.



(a) Roads with MAX SPEED > 100 km/h



(b) Roads with 60 km/h < MAX SPEED < 100 km/h



(c) Roads with MAX SPEED < 60 km/h

Figure 16. Spanish regulation regarding discontinuous lines' size [44].

3.1.3.2 *Lane width*

The width of the road lanes varies with the type of lane under analysis. These dimensions, unlike the case of discontinuous lines, are less restrictive, and allow a bigger error [45]. However, as we will see later in Section 3.3, this will not pose a problem to the implementation of the system. The width of each lane depending on the typo of road is shown in Table 2.

Type of road	Width
Highways / Freeways	3.75 m
Conventional roads	3.50 m
Access ways	3.00 m
Mountain roads	
Roadways and urban roads	
Urban roads with more than 2 lanes per direction	2.75 m
Urban roads with less than 200 households	2.50 m

Table 2. Lane width for each type of road [46].

3.2 System overview

The objective of the proposed solution is to identify certain driving behaviours. The focus of this work has been tailgating detection. As a result, the modules have been optimized to detect this specific behaviour. With the objective of achieving proper detection, we attempt to obtain geometric models that define the analysed scene. This way, the problem is characterized and enough data is expected to be obtained in order to yield a conclusion on whether the analysed scenario presents the specified behaviour or not.

The proposed system, depicted in Fig. 17, is based on the analysis of a video flow. In this section, we present a brief explanation of the whole system, and a deepened explanation of each individual module will be provided in the following section (3.3).

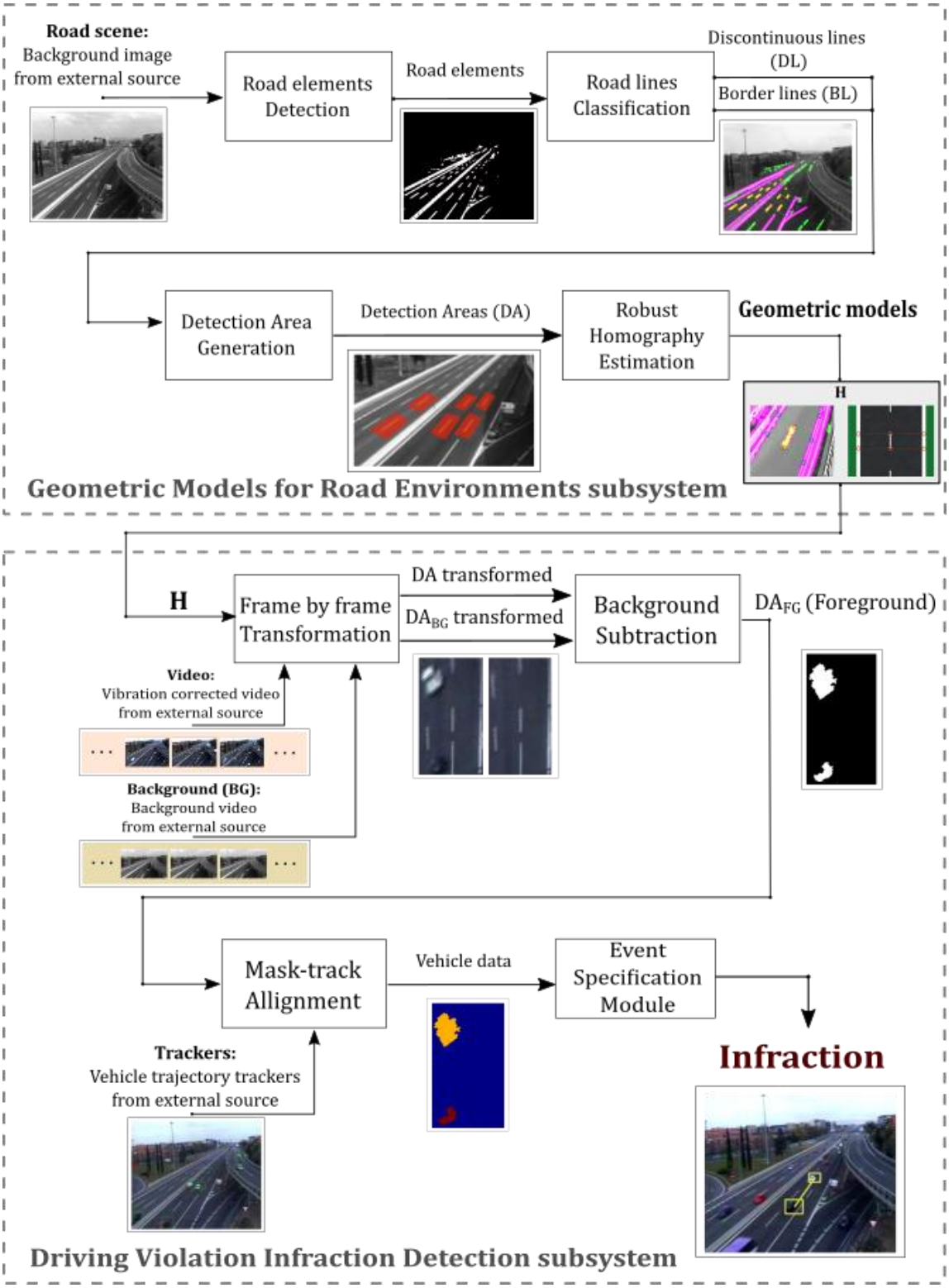


Figure 17. General pipeline of the proposed system to detect tailgating behaviour.

As it is shown in Fig. 17, the system is divided into two main subsystems. These subsystems are the following:

- Geometric Models for Road Environments subsystem → GMRE subsystem.
- Driving Violation Infraction Detection subsystem → DVID subsystem.

The first subsystem (GMRE) is intended to obtain the geometric models of the road. For this, the input used is a background image of the scene to be analysed. This background image is retrieved from the output provided by the external previous system (see Section 2.4). The first step of the process is to identify the road elements, namely, the white marks present in the road. Then, we need to classify the obtained elements. We do so into two categories:

- **Discontinuous lines:** The discontinuous lines (henceforth referred to as ***DL***) are the discontinuous marks that separate each lane in the same direction within the road.
- **Continuous/Border lines:** The border lines (henceforth referred to as ***BL***) are the lines that define the borders of each road. In some cases, they can also separate lanes.

Once this is done, the next step is to identify some key points to eventually estimate a homography matrix that, given the original view of the road, transforms it into a normalized top-view. For this, we first need to discriminate the areas where we are going to be able to take measures. These areas will be henceforth referred to as Detection Areas (DA). Within these DAs, we need to identify points from the image belonging to elements of the road for which the dimensions are known, and so a model has been defined to include those elements (see Section 3.1.3 for regulations). After the points of interest are set, we use this set of points to estimate the homography matrix that will define the geometric models. This process is performed in the *Robust Homography Estimation* module.

These models, along with the original video and the updated background video of the scene, will be the input to the second subsystem (DVID). The first step of this second subsystem will be to perform a frame by frame transformation to both input videos using the geometric models obtained in the GMRE. This will yield the DA corresponding to the top views of the scene. As we have the top view of the original video and the top view of the background, we perform a background subtraction to detect the foreground moving elements in the DA (the vehicles). This DA is then aligned with the tracking information that we have on the vehicles from the system in 2.4. Once each object of the image has been associated with a vehicle, measures are taken and we try to detect the corresponding behaviour in the last module, which will produce as an outcome a list of the infractions that occur in the scene.

The stage devoted to the design of the system was particularly important for this project, as it concerns two different subsystems, and so the conditions for both systems to properly match and exchange the necessary information correctly. After the design of the system was completed, the implementation followed. Through the next two sections, we will present a detailed explanation of the functionality thought for each module displayed in Fig. 17 during the design stage. In order to facilitate the understanding of the project for the reader, any relevant comments or explanations regarding the implementation will be also commented as we present the functionality of each module.

3.3 Geometric Models for Road Environments subsystem (GMRE)

The first subsystem is intended to define the scene. In order to do so, we attempt to retrieve a transformation that models the original perspective of the image into a view in which we can identify events and take measures. For this, we need to have accurate knowledge about at least some of the elements of the road. Therefore, this is precisely the starting line of the project.

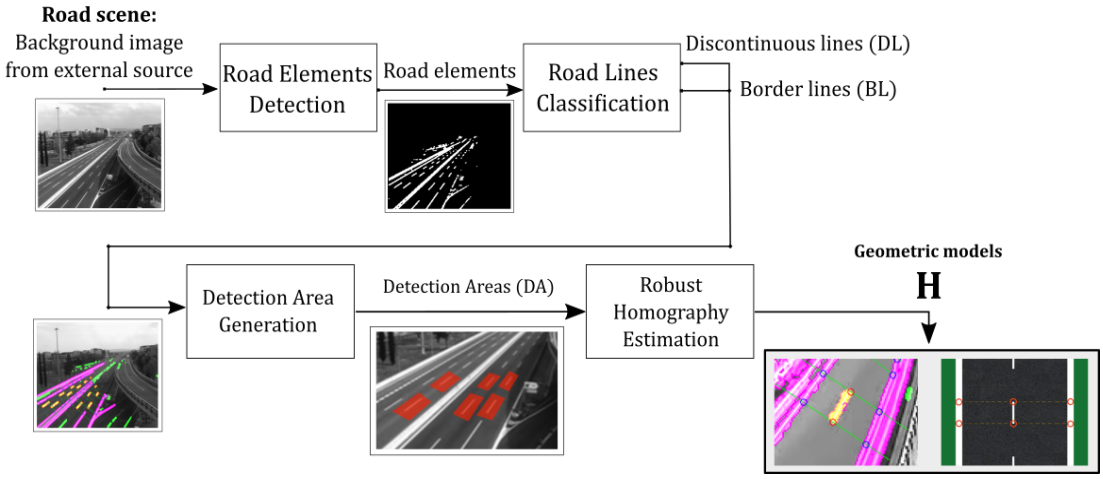


Figure 18. GMRE subsystem general pipeline.

As it can be observed in Fig. 18, this first subsystem has four main modules. In the following sections, we will explain each module.

3.3.1 Road elements detection

The elements chosen to define the road are the white line marks and the different lanes that shape the road. Therefore, the first module was designed as a first filter that discards those elements of the image that are not relevant to the system.

It is worth mentioning that this subsystem does not work with the complete video flow, but rather with a steady image extracted from the video under analysis through the system defined in 2.4. This image represents the background of the scene during the period of the video. Some examples of background images are shown in Fig. 19. In the mentioned figure, it can be observed that the traffic scenes used during the development of this project present varied camera locations with different characteristics and features. Some of the scenes present challenging obstacles that we will attempt to overcome.

Figure 19. Examples of background images used as input of GMRE.

This background image is, then, the first input of the GMRE subsystem (Fig. 20.a). In this image, this first module attempts to identify the elements of the road. This is done through an edge filter.

This filter, however, is adapted to obtain better performance in this particular situation, so we will not use an established method (Prewitt, Canny, Laplacian of Gaussian, etc.) but rather a specific filter design solely for the purpose of detecting white line road marks.

The system described in 2.4 provides us with a road mask obtained by analysing the trajectories of the vehicles in the scenes (Fig. 20.b). This mask is used to directly discard those parts of the image in which vehicles are strongly unlikely to appear. However, we perform dilation in the mask in order to have a margin to detect those road marks that may not be part of the mask (Fig. 20.c).

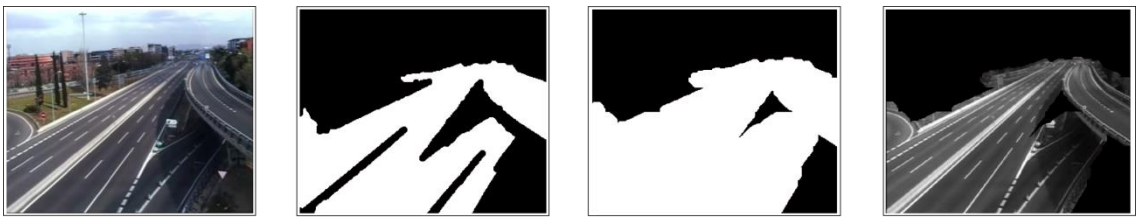


Figure 20. Mask dilation process.

An intersection operation is then performed with the dilated mask and the background image, resulting in an image like the one shown in Fig. 20.d.

In order to detect the road marks, we calculate the gradient magnitude and the gradient orientation of the image (neglecting the parts of the image corresponding to the edges of the dilated mask). As the gradients due to the road lines are dominant in the road area, and

assuming that the orientation of the elements of the image corresponding to the road marks must be similar (although not parallel due to the camera viewpoint), we give more weight to those gradient orientations that have greater accumulated magnitude. In other words, any object in the image is more likely to be identified as a road mark if its orientation is similar to that of the majority of objects present in the scene.

To illustrate this process, Fig. 21 displays a weighted histogram *H(o)* showing the most common orientations of the objects present in a background image (Fig. 20.a).

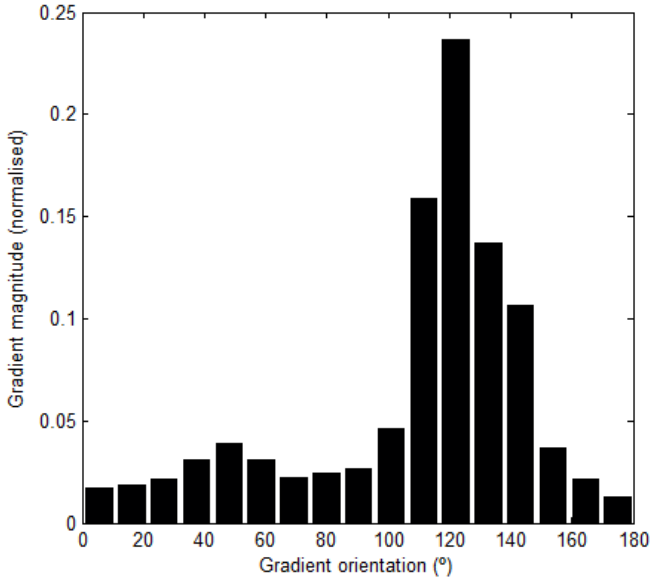


Figure 21. Weighted histogram *H* showing the most common gradient orientations for a particular scene.

We can see that there exists a predominant orientation in the image, which corresponds to that of the road marks. Taking this into account, we proceed to calculate an acceptable interval around this main orientation. For this, we define a required magnitude value. Each magnitude orientation will have to satisfy this value to be considered as a part of the interval. We define this value as:

$$M_{req} = f_o \cdot \max(H) \tag{9}$$

where M_{req} is the required value, H is the weighted histogram containing the magnitude of the gradient, and $0 \leq f_o \leq 1$ is the factor for which we multiply the maximum value of the histogram to obtain the required value.

From (9), we establish a strict orientation vector S_o:

$$S_o(o) = \begin{cases} 0, & H(o) < M_{req} \\ 1, & H(o) \geq M_{req} \end{cases} \tag{10}$$

where *o* is the index variable for the bins representing the orientations (in the range of 0º-180º). This vector is set to 1 for those orientations for which the original histogram is large enough.

Then, this strict orientation interval is widened to allow some tolerance, determining the final vector of valid orientations (I_o):

$$I_o(o) = \begin{cases} 1, & S_o(o) + T = 1 \\ 1, & S_o(o) - T = 1 \\ 0, & otherwise \end{cases} \quad (11)$$

where T is the tolerance factor in degrees. These parameters were heuristically adjusted, resulting in the following optimal values:

- $f_o = 0.65$
- $T = 20\ degrees$

Hence, the elements in the final vector I_o that are set to one correspond with orientations that will be considered in the subsequent analysis, whereas those orientations with null values will be discarded.

Summarizing, we first have a gradient magnitude image (Fig. 22.a). Then, we apply the orientation filter to this gradient image, resulting in a binary image with the edges of the road marks (Fig. 22.b), where the elements have been discriminated according to their orientations. However, as the inner part of each road mark does not have edges, empty spaces that are actually corresponding to the road marks can be observed. To correct this, we perform morphological image processing. More specifically, we perform a closing operation (see 2.3.3), which yields the image in Fig. 22.c.

(a) Gradient magnitude image (b) Binary edge image (c) Edge image after closing

Figure 22. Edge detection process.

To put an end to this module, a soft opening operation (see 2.3.3) is performed on the image to discard small imperfections, resulting in what will be considered for the next module as the image containing the road elements. The output of this first module is displayed in Fig. 23.

Figure 23. Road element detection output.

3.3.2 Road lines classification

In the first module the system identified the elements of the image that can be considered as road elements, these are, the white line road marks. In order to properly define a model of the road, we have divided the elements of the road into two different categories. As it was mentioned in 3.2, these are the *Discontinuous Lines (DL)* and the *Border Lines (BL)*.

One of the main objectives during the development of this project was to obtain a system as autonomous as possible, but still keeping an acceptable performance to get usable results. For this module, two lines of research were launched, deriving in two different methods that discriminate the DLs and the BLs from the road elements provided by the *road element detection* module. The two methods developed are the *automatic method* and the *semi-automatic method*. Both methods work with the same input and provide equivalent output, so either of them can be used and no changes are required in the configuration of the rest of the system.

3.3.2.1 *Automatic method*

This method does not require any human interaction at all. The method attempts to discriminate the DLs and the BLs based on their characteristics.

3.3.2.1.1 Detecting the DLs:

First, the DLs are discriminated. For this, we obtain the following characteristics of every object present in the image (see *regionprops*[1]):

- Eccentricity
- Orientation
- Centroid
- Extrema
- Length (distance between the farthest pair of extrema)

Every object in the image becomes now a candidate to be considered as a DL. This section describes the filters that have to be obeyed by a candidate in order to be finally labelled as DL.

Eccentricity filter: The first filter models the shape of a candidate. The previous module established a very basic concept of what it is considered as a road element in order to allow certain tolerance for this next step. However, we can now perform certain operations to narrow the concept of line a little more. We know that the shape of a road mark is a straight line; therefore, a candidate must have a high eccentricity to be considered a DL. Defining the eccentricity range as (0-1), with 0 being a perfect circle and 1 being a straight line, the required eccentricity of the object for further processing is 0.96.

Length filter: A candidate must be large enough to be considered a line, but not too large, as we know that DLs are shorter than BLs. We assume to have at least one BL in the image, and we assume that said BL is significantly larger than any DL; therefore, the length of the candidate must be less than half of the length corresponding to the largest element of the image to be considered for further processing, and also more than 1/20 of said length. This is:

$$\frac{1}{20} \cdot L_{max} \quad < \quad L \quad < \quad \frac{1}{2} \cdot L_{max} \tag{12}$$

[1] MATLAB function *regionprops*: http://es.mathworks.com/help/images/ref/regionprops.html

where L_{max} is the length of the largest element in the scene, and L is the length of the object under analysis.

Contiguity filter: A DL is assumed to be contiguous to at least one other road line. To express contiguity between objects, we define each object by its centroid and its orientation. An element e_2 is considered to be contiguous with respect to another element e_1 when the following two conditions are satisfied:

1. Its centroid c_2 is closer than a given distance d_c to the straight line formed by the centroid of the other element c_1 and its orientation angle α_c.
2. Its orientation angle β_c differs no more than a given limit δ_c from α_c.

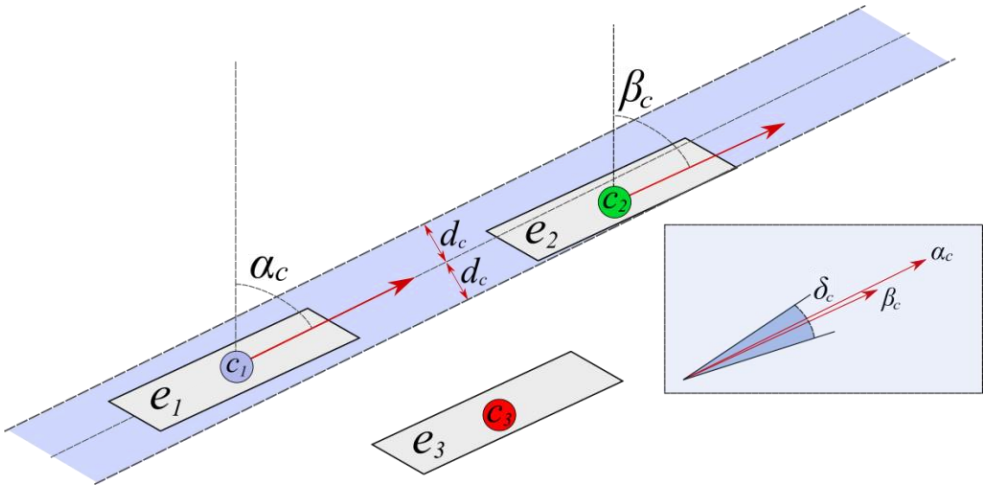


Figure 24. Contiguity concept definition.

In Fig. 24, we can observe how element e_2 is contiguous to element e_1, as centroid c_2 lies within the acceptable distance range from the orientation line of e_1, and their orientations are sufficiently similar. On the other hand, one can note that e_3 is not contiguous to e_1, as its centroid lies far from the acceptable distance range, given by d_c.

The optimal values for these parameters have been heuristically computed, resulting in the following:

- $d_c = 3\ pxl$
- $\delta_c = 6\ degrees$

Distance filter: We have seen that the previous module provides this one with an image containing the elements of the road. Nevertheless, this binary image is often noisy due to low image resolution. Therefore, we must be careful with elements that might have been detected but are not actually road marks contaminating the system. Discontinuous lines are doubtlessly contiguous to other road marks, but there might be other elements in the scene that are also contiguous. This is the reason why candidates must successfully pass one last filter, which analyses the distance between elements.

Maximum and minimum gaps between elements are defined. Each element must satisfy these limits to be considered a DL. The limits are defined based on the length corresponding to the element under analysis. Both limits (minimum and maximum) are a factor of the line´s length:

$$g_{min} = \gamma_{min} \cdot L\,, \quad 0 < \gamma_{min} < 1 \tag{13}$$

$$g_{max} = \gamma_{max} \cdot L\,, \quad \gamma_{max} > 1 \tag{14}$$

where g_{max} and g_{min} are the maximum and minimum gap limits, γ_{max} and γ_{min} are the factor and L is here the length of the analysed element. Again, we have heuristically computed the values of the limiting factors:

- $\gamma_{max} = 2.75$
- $\gamma_{min} = 0.7$

The *distance filter* must be satisfied for every element in the scene, including those that were declared non-contiguous to the analysed candidate. This means that, in order for the candidate to be labelled as DL, there is an area around it in which no elements must have been detected (shaded red in Fig. 25); after the threshold determined by g_{min}, there is an area in which only contiguous elements can assert its DL condition (this area is shaded blue in Fig. 25). This area ends with the upper threshold (determined by g_{max}), after which lies the outer area (shaded yellow in Fig. 25), in which not even contiguous elements can assert the DL label to the analysed candidate. Nonetheless, elements found in this outer area do not impede the candidate to be labelled as DL.

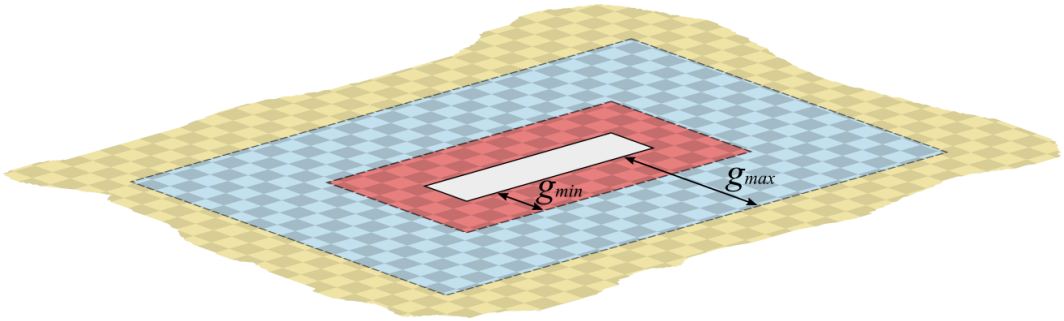


Figure 25. Distance filter illustration.

In order to gain robustness, this module adds a special *parity condition* to the analysed candidates. This condition is that, in order for the analysed candidate e_1 to be considered as a DL, the element $e_2$that makes e_1 satisfy the before mentioned conditions must also satisfy them. This means that the system will detect either zero or at least two DLs in the scene, but never just one. An example of this condition is displayed in Fig. 27, where e_1 and e_2 are successfully identified as DLs. This condition helps the system discard false DLs caused by the *intermittent border line effect*. Such effect occurs when a road mark that can be visually identified as a BL (see Section 3.2 for brief definition) is not detected as a whole, but rather in pieces (Fig. 26). This can happen for various reasons, such as poor image resolution or brightness intensity (see 3.1.2.2 for restrictions). In Fig. 27, it can be observed how e_3 and e_4 are actually parts of the same BL in the scene, but are here detected as two different objects. The *parity condition* effectively discarded e_3 as a DL even though its eccentricity, length, orientation and contiguity

with another element of the scene are valid, because the element that caused for it to satisfy the conditions (e_4) does not satisfy them itself.

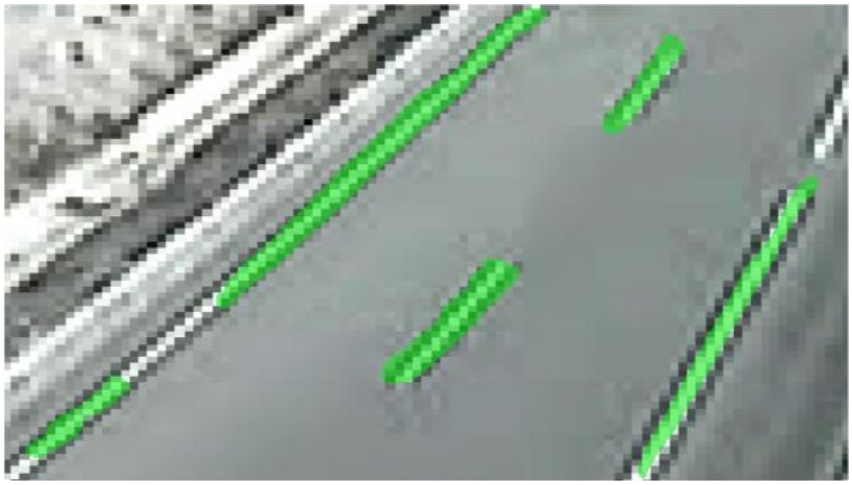


Figure 26. Intermittent border line effect.

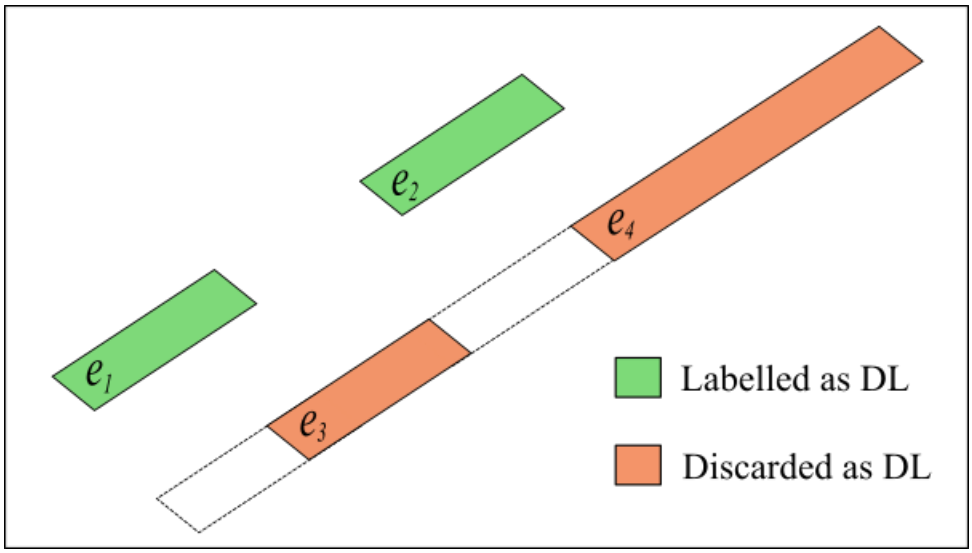


Figure 27. Parity condition effect.

3.3.2.1.2 Detecting the BLs:

After this, we know which elements of the scene are DLs. However, we still have to identify the BLs present at the scene. The method to do so is simpler than the one we just explained; as we only have two obtain three characteristics from the set of road elements (see *regionprops*[2]):

- Eccentricity
- Extrema
- Length (distance between the farthest pair of extrema)

The number of filters a candidate must pass to be labelled as BL is lower and the filters are much simpler, as we are looking for a fairly distinguishable element. Continuous or border lines

[2] MATLAB function *regionprops*: http://es.mathworks.com/help/images/ref/regionprops.html

are characterized for being larger than discontinuous lines; as such, the detected elements that are ideally going to be labelled as BL are expected to be extremely eccentric and possess a great length.

Therefore, the method for discriminating BLs consists solely on two filters, a length filter and an eccentricity filter. It is trivial to deduce that one determined element cannot be labelled as DL and as BL, but we will see that this is impossible due to the filters' own nature.

Length filter: A candidate must satisfy a minimum length. We saw that during the DL discrimination process a length filter was also used, and a maximum length limit was applied to any candidate. This limit depended on the length of the largest element of the road (L_{max}). In this case, the filter specifies a minimum length that a candidate must satisfy, also depending on L_{max}. This can be expressed as:

$$L > \frac{3}{4} \cdot L_{max} \tag{15}$$

From (12), we can deduce that a candidate can never be labelled both as a DL and a BL.

Eccentricity filter: This filter works exactly the same way as its analogue in the DL discrimination process. However, as BLs are longer, an even higher eccentricity is expected; therefore, the minimum eccentricity required to be labelled as a BL is 0.99.

Using these two simple filters we are able to discriminate the BLs from the rest of the elements provided by the previous module. Once we have the DLs and the BLs properly identified we can proceed to apply further processing. The set of discontinuous and border lines is, hence, the output of the road *lines classification module*.

3.3.2.2 *Semi-automatic method*

The semi-automatic method provides an alternative to the method explained above, in which a little user interaction is required. The objective of this method is to provide us with a more reliable, human supervised output for this module. The performance of both methods will not be compared until Chapter 4.

The idea of this method is very simple. Given a pre-selection of the road elements provided by the system's first module, the user will be given the chance to select which of those elements are DLs and which are BLs. For this, a visual output of the *road elements detection* module is presented on a screen (Fig. 28).

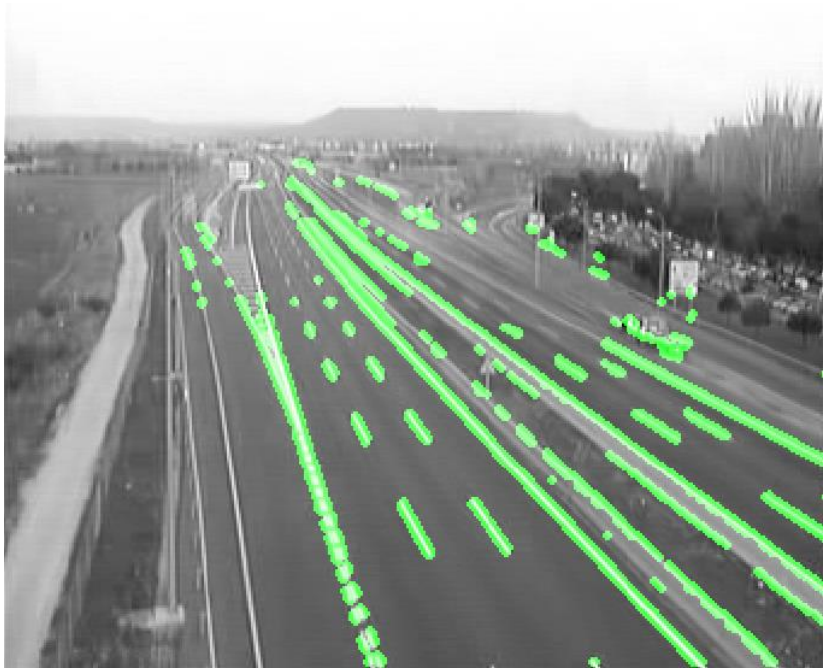


Figure 28. Pre-selection of road elements provided by previous module.

Through an intuitive interface, the user is first asked to select the DLs out of the lines emphasized in green just by clicking on them. This results in a screen showing the user's selection (Fig. 29), and requesting a confirmation in case there are any mistakes.

Figure 29. DL user selection (yellow).

Once the user has confirmed this selection, the program asks the user to select, from the rest of the road elements, those that correspond to BLs (using the same procedure as in the previous step). It is worth mentioning that this method allows the user to identify certain BL that might

not be detected by the automatic method. After clicking on the BLs, the program will show the selection again to the user, requesting for a new confirmation (Fig. 30).

Figure 30. DL user selection (yellow) and BL user selection (purple).

The system will align the selected points on the image with the corresponding elements of the scene, and effectively store the set of DLs and BLs selected by the user. This data is then used as the input for the next module of this first block.

3.3.3 Detection Area generation

This module is devoted to extract the Detection Areas (DA) from the Discontinuous Lines (DL). The process is very simple: it identifies the different DLs present on the image and it separates them into different DAs.

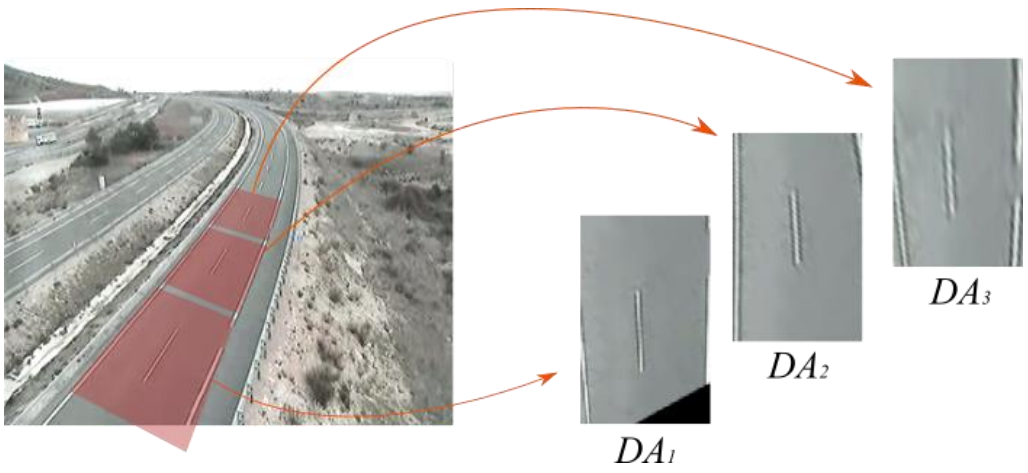


Figure 31. Generation of Detection Areas (left) and their corresponding transformed top-views (right).

The objective is to extract a top-view DA from every DA identified in this module, like it is shown in Fig. 31. For this, we need to establish several homography matrices that are able to transform from the original domain to each of the transformed DAs. This will be done in the next module.

3.3.4 Robust homography estimation

As it is briefly stated in Section 3.2, the geometric models of the road are to be obtained through a homography matrix. This matrix models any kind of transformation of a planar element, which fits well with the road surfaces. In order to estimate this homography, we need to be able to identify a set of points for which the coordinates in both domains are known. However, we will not be implementing a transformation for the scene as a whole, but rather several transformations, each one referred to a particular Detection Area (DA).

Each DA will be centred on a DL. Thus, it is easy to deduce that there cannot be more DAs than detected DLs. In order to estimate the homography for a given DA, we establish a set of points whose coordinates are known in the transformed domain, which corresponds with a normalised top-view of the scene. For this module, a set of six known points in the top-view is defined, and the objective is to find the equivalent points from the original view to be able to map them. This will be henceforth referred to as the *six-point model (SPM)*.

3.3.4.1 *Key points identification: Six-point model (SPM)*

In order to determine a homography, the DLT method requires a set of at least 4 points whose coordinates are known in both domains. As we stated before, this system implements a six-point model. There are two main reasons for choosing six points for the model. The first reason is that it adds robustness to the algorithm, allowing for some of the points to be noisy and still getting a fair result. The second is related to the particular nature of the analysed scene, which can be defined by two points for the central DL and two more points at each side of the DL representing two lanes of the road.

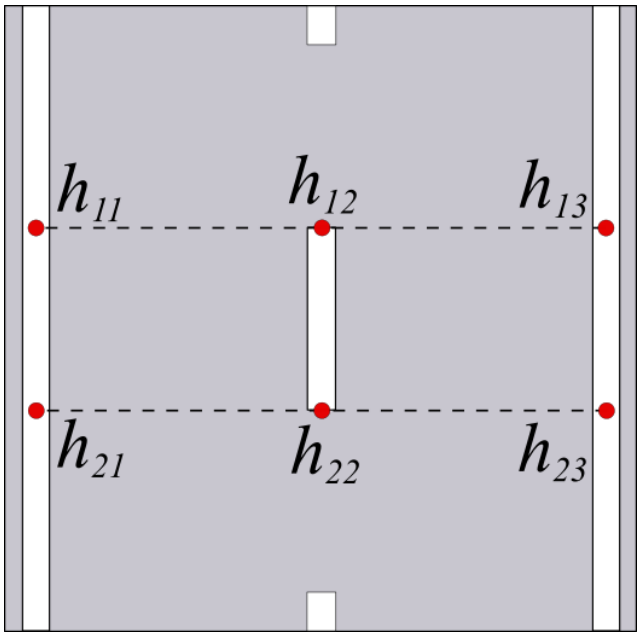


Figure 32. Six-point model scheme.

The central element of each DA is the DL. Therefore, there are two fixed points for the model, corresponding to the top and bottom extrema of the DL. In Fig. 32, these two points are named h_{12} and h_{22}. From these two points, two straight lines are thrown orthogonally to the DL's orientation. Intersections with BLs (and other DLs in the case of multi-lane motorways) in the scene are then considered for the other four points in the SPM. However, it is noteworthy that not all of them have to be available to compute the transformation, thus providing a robust solution against visual artifacts.

There are two methods two obtain the SPM, the *standard lane method* and the *multi-lane method*. The system automatically decides which method to use depending on the geometric layout of the DA.

The points belonging to the central DL (h_{12} and h_{22}) are calculated the same way in both methods. This is done by simply obtaining the extrema of the object identified as the DL and selecting the pair of extrema that are farthest apart.

Standard lane method: This method is used for road scenes where there are only two lanes separated by a set of DL (Fig. 32). In this case, the method consists solely on what was explained above: two sets of orthogonal lines are thrown from h_{12} and h_{22} and the intersections with the BLs are calculated.

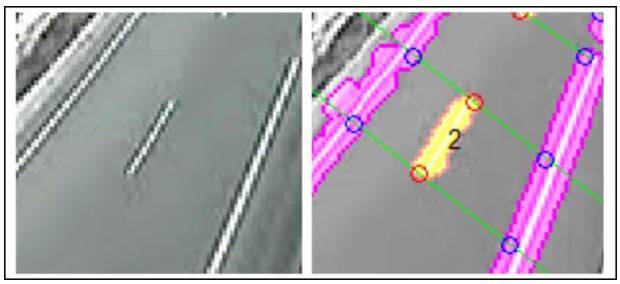


Figure 33. Standard lane scenario.

The lines are orthogonal with respect to the orientation of the DL. Thus, we would like to note that our approach is just an approximation and would be perfect only if the central DL and the nearby lines were parallel which, in general, does not completely hold due to the projective view of the scene. However, it is well known that for planar surfaces (as the road) of small area with respect to their distance with the camera centre, projective transformations can be successfully approximated by affine transformations which do not break the parallelism between lines. Under this hypothesis, the use of orthogonal lines from the DL to the nearby lines will be precise enough.

Multi-lane method: This method is used for multi-lane motorways with more than one DL separation (Fig. 34). As it is still assumed that the approximation of projective transformations by affine transformations is valid, the method for finding nearby objects is the same as in the previous method. However, as in this case there are more lanes, the orthogonal lines can intersect with DLs as well as with BLs.

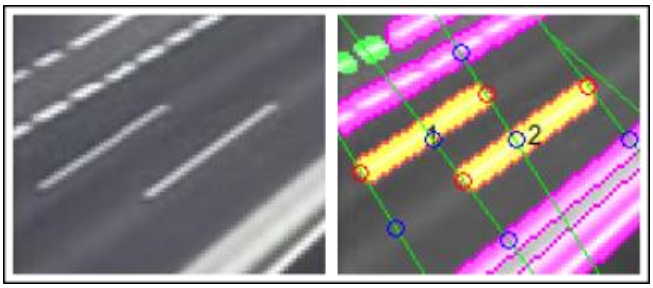


Figure 34. Multi-lane scenario.

It often occurs that one of the orthogonal lines intersects with a nearby DL, while other does not encounter an intersection point until it reaches a BL farther apart. This effect can be observed in Fig. 35.a, where the DL marked as 1 throws an orthogonal line from the top point that intersects with DL 2, while the line from the bottom part does not encounter a DL from the same separation, and rather intersects with the BL in the next lane. However, the multi-lane method detects this situation, and it is able to obtain the correct point that matches the model, as it is shown in Fig. 35.c.

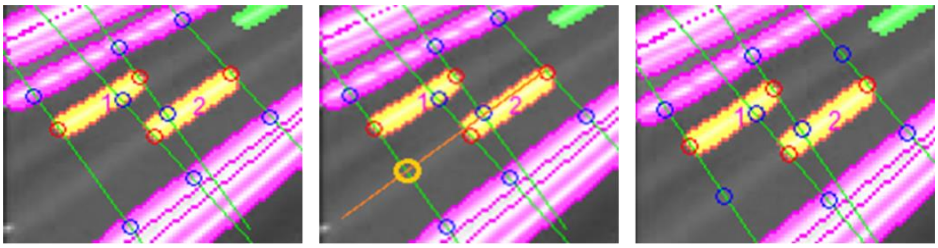


Figure 35. Multi-lane correction.

The correction is performed if (as it happens with DL 1 in Fig. 35.a) the distance from one of the intersection points to its corresponding DL point ($h_{12} \rightarrow h_{13}$) differs significantly from the distance between the analogue pair of points ($h_{22} \rightarrow h_{23}$). When this happens, the point that is farthest apart (in this case, h_{23}) is discarded, as we assume an intersection with a neighbouring DL to be more trustworthy. The system attempts then to find the correct location of this point. For this, it assumes that the first intersection point (h_{13}) is correct. From this point a straight line is thrown with the same orientation as the neighbouring DL (in this case, DL 2). The corrected point will be the result of the intersection of this line with the orthogonal line that first expected to find it (marked in yellow in Fig. 35.b). This way, we obtain the correct point (h_{33}) even when there is no intersection element.

3.3.4.2 *An algorithm for Homography Robust Estimation*

Once the key points have been identified, we can use them to estimate a homography matrix H. This matrix will be used to transform the original view into a normalised top-view (Fig. 36).

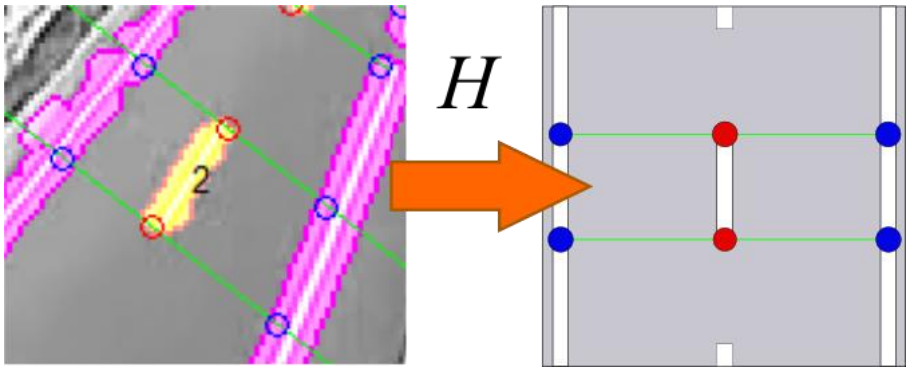


Figure 36. Diagram illustrating ideal transformation from original to top-view.

However, some of the detected points might not be accurate enough, causing the system to unnecessarily decrease its performance. In these cases the transformation would improve just by discarding these incorrect key points. Consequently, we have developed a method to add robustness to the point selection. The system, inspired by the RANSAC (RANdom SAmple Consensus) method [38], has been specifically designed for our particular scenario of a DA. The method here developed is not random, but it chooses the optimal combination of detected points in order to obtain the highest possible quality for the homography. Thanks to this method, we are able to discard possible outliers that may affect negatively to the resulting transform.

We need to find at least four mapped points in order to be able to estimate a homography, but these points cannot be a linearly dependent or the system will be underdetermined. As some combinations include points that are linear combinations of others, they cannot be used. In the case of finding 5 or 6 points, there is no problem with this matter. However, when only 4 usable points are found, invalid homography matrices can be obtained. Table 3 shows the combinations of points that result in linear dependency (the notation from Fig. 32 is used to determine the points of the SPM). Note that h_{12} and h_{22} are always present in the calculations, as they define the DL; therefore, the table only represents the different pair combinations of the other four points.

	h_{21}	h_{13}	h_{23}
h_{11}	✓	✗	✓
h_{21}		✓	✗
h_{13}			✓

Table 3. Four point combinations.

Note that the invalid combinations are those that include, apart from the points in the DL, the other two points in one of the rows and neither of the points from the other row. If any of the invalid combinations occurs, the DA is marked as unusable and is discarded.

Before going into detail with the method for robust homography estimation, let us remember that the method chosen to estimate homographies is the Direct Linear Transformation (DLT).

Deepened information about this method can be checked in [37]. Moreover, the process of the homography has been reviewed in this manuscript in Section 2.3.1.

If the number of valid points is 4, and the set is valid (mapped points are not linear combinations), our system directly computes the homography using the DLT algorithm. However, if the combination is valid and contains more than four points, we then run our robust approach for the estimation. Our algorithm for robust estimation operates as follows: it takes, out of the set of points, every possible (and valid) four point combination and calculates the homography matrix *H*. It then transforms (using the obtained *H*) the points that were not selected in this combination. An error is calculated based on the Euclidean distance between the transformed point and the ideal output of such point. If this error is below a certain threshold, it is considered as an *inlier*. If it is not, it is considered as an *outlier*. After repeating the same operation with all the combinations, the transformation matrix with the greater number of inliers is the chosen as the correct one.

Example: In the case represented by Fig. 37, we have a DL for which we have found five out of six possible points in the model. These are (according to the notation in Fig. 32):

$$h_{12}, h_{13}, h_{21}, h_{22}, h_{23}$$

In order to improve the performance, we apply the robustness algorithm to the set of points. As h_{12} *and* h_{22} *are fixed points, there are 2 possible and valid combinations from the other points:*

$$h_{12}, h_{13}, h_{21}, h_{22} \quad \& \quad h_{12}, h_{13}, h_{22}, h_{23}$$

Homography matrices H_1 *and* H_2 *are calculated from the previous sets of points (respectively). When calculating the inliers of the matrices it yields that* H_1 *has one outlier, as the transformation is not completely accurate, whereas all points are considered inliers when applying* H_2*.Therefore,* H_2 *is selected as the homography transformation.*

If the number of inliers resulted to be the same, then the transformation matrix calculated with the complete set of 5 points would be selected.

The same reasoning follows when obtaining the best combination from a set of 6 points.

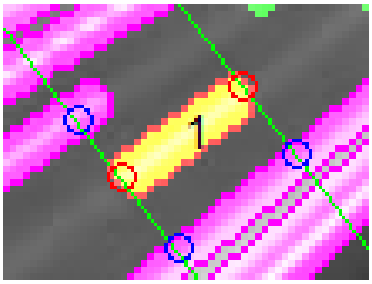


Figure 37. DL with 5 out of the 6 possible points found.

Note that the output of this module will be the homography matrix that will transform the original view of the image into the normalized view with which we are able to work. In our case, the ideal view is a top-view in which measures can be taken. Furthermore, a top-view facilitates the detection of rather important features like lane identification or occlusion cases.

As a conclusion for this module, Fig. 37 shows a diagram of the functioning of the explained algorithm.

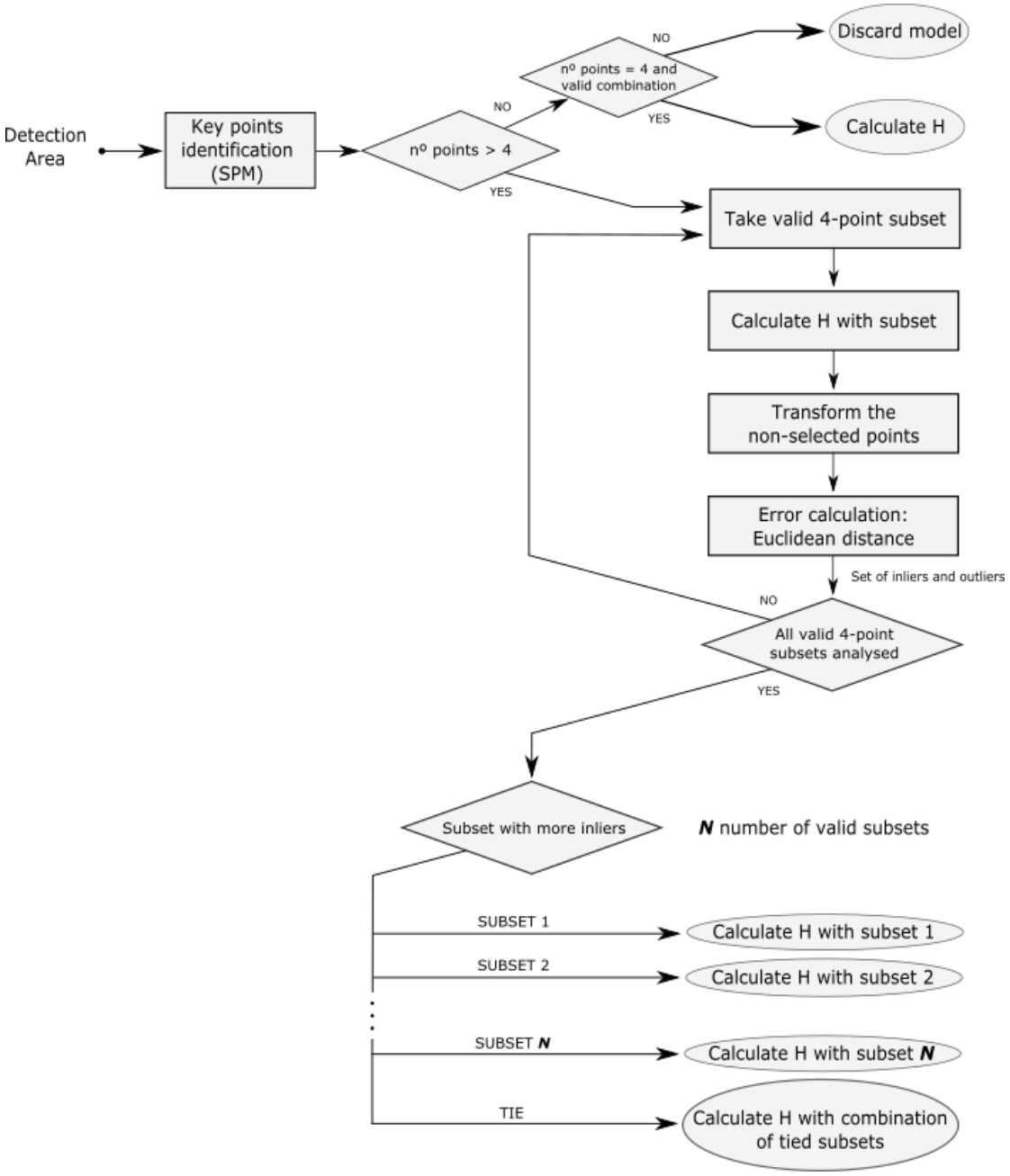


Figure 38. Diagram showing the algorithm for homography robust estimation.

This last module provides us with the output of the first subsystem (GMRE), which is the set of homography matrices that provide the transformed top-view of each DA, like it is displayed in Fig. 39. This output will be used in the second subsystem to retrieve the characteristics of the scene's video flow.

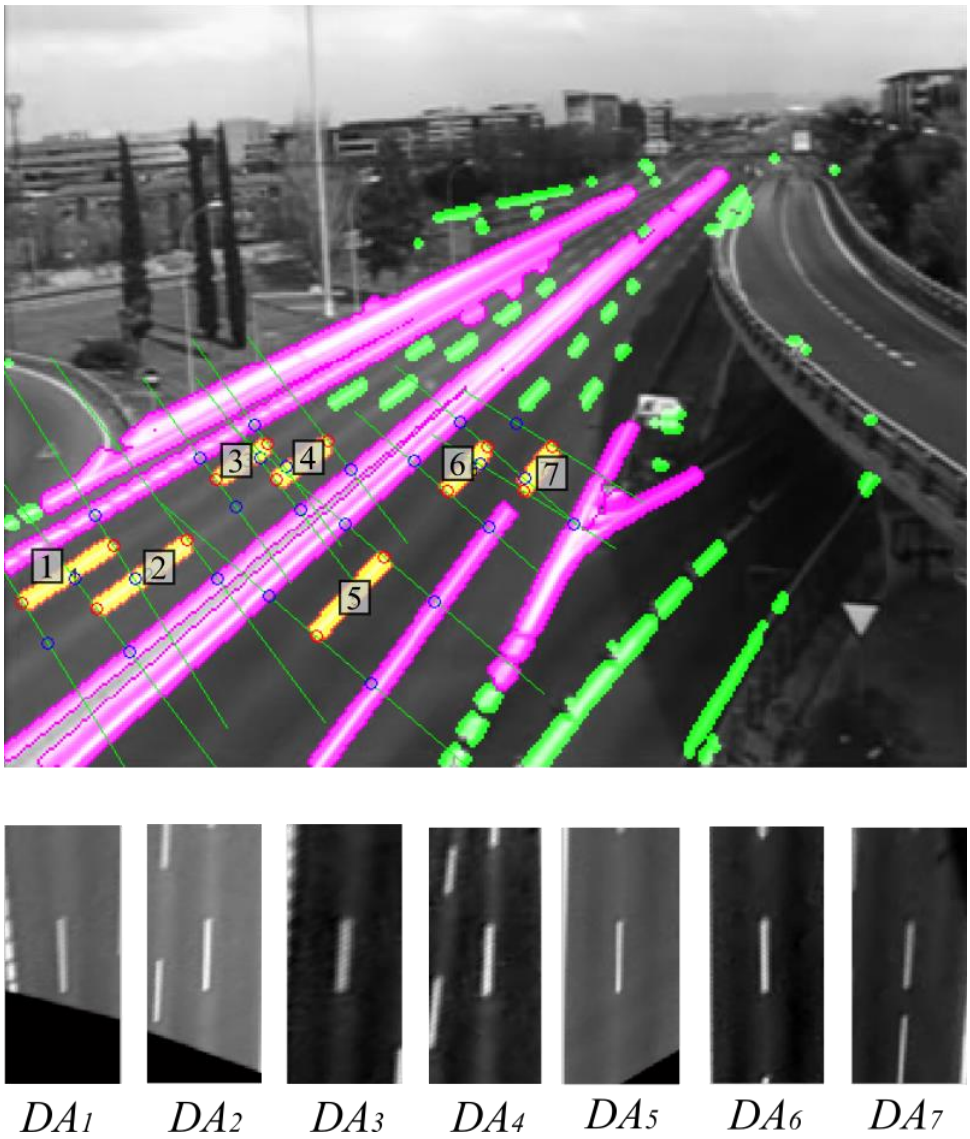


Figure 39. Set of transformed DA for a certain scene.

3.4 Driving Violation Infraction Detection subsystem: Tailgating (DVID)

The second subsystem if composed of yet another four modules, as it is shown in Fig. 40, and it is in charge of providing the final output of the system: tailgating detection.

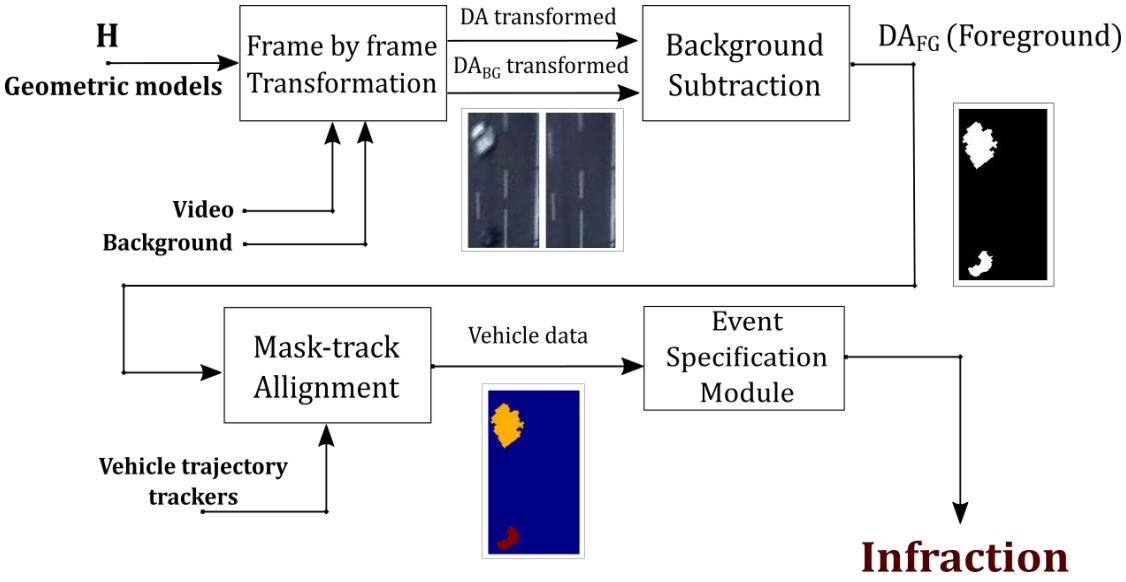


Figure 40. DVID subsystem general pipeline.

It should be noted that, although computationally speaking this second subsystem is heavier than the first one, conceptually it is much simpler. In the following sections we will explain the functionality of each of the four modules.

3.4.1 Frame by frame transformation

As opposed to the GMRE subsystem, this second block is fed with the video flow of the corresponding traffic scene. Therefore, a frame by frame analysis is carried out. Moreover, the output from the previous subsystem is directly fed to the first module of the DVID subsystem in order to work with the convenient top-view DA.

This first module is intended to perform a transformation of the corresponding frame. Expectedly, the transformation that will be used is the output of the GMRE subsystem. As we have explained in Section 3.3.3, there can be more than one transformation matrix in one scene, one associated to each of the detected DAs. This set of matrices, henceforth denoted as H_t, is applied to the frame under analysis to generate the corresponding top views of the detected DA. The current frame is extracted from two different video flows, as it is shown in Fig. 40. The first video represents the scene as is, with the vehicles currently in the scene (I_o). The second video represents a background model at the same moment (I_{bg}). From this, it is trivial to deduce that we will obtain two sets of DAs, one set with vehicles (DA_o) and the other containing only the background (DA_{bg}).

Figure 41. Detected DA in a scene (left). For a given frame in the video, normalized top-view of the DA (middle) and the background DA (right).

The output of this module is then the two sets of DA_o and DA_{bg}. In Fig. 41 we can see an example of a particular element of both sets: the middle image represents a DA and the right-hand image represents is corresponding background DA_{bg}. It is worth mentioning that these normalised views are a key step in the process. This is because in these views we know exactly the dimensions of the elements of interest (lanes, lines, etc.), which allow us to take quantitative measurements (vehicle locations, distance, speed, etc.).

3.4.2 Background subtraction

This simple module implements a subtraction between the sets of images obtained as an output of the *frame by frame transformation module*. The objective is to have an idea of the vehicles that are passing through the analysed DA. This can be expressed as:

$$DA_{fg} = DA_o - DA_{bg} \tag{16}$$

where DA_{fg}, represents the set of foreground DAs.

A threshold is then applied to the resulting image, obtaining a binary representation of the foreground. After a soft post-processing stage in which morphological processing is applied to get a smoother foreground image, we obtain a set of images that are similar to the one displayed in Fig. 42.

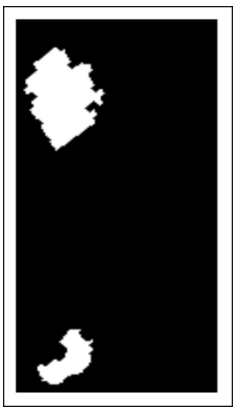


Figure 42. Foreground DA corresponding to Figure 41.

The set of foreground Detection Areas (DA_{fg}) is the output for this module.

3.4.3 Mask-track alignment

This module is devoted to align foreground masks with the tracked vehicles. Up to now, we have been able to identify the dynamic objects of each DA via a background subtraction process. It is assumed that these dynamic objects correspond to the vehicles passing through the scene at the analysed frame. However, if we want to take measures and associate them to individual vehicles, we need to establish such concept.

Consequently, another input to the system has to be considered here. The previous system introduced in Section 2.4 tracks individual vehicles in the scene and provides their trajectories. Hence, the tracking system provides the trajectory of every vehicle passing through the scene as a sequence of spatial locations and shapes (encoded as bounding boxes) (Fig. 43). Hence, we can get the bounding boxes associated with every vehicle at every frame. Although one might think that the output of the tracking system is enough to support the tailgating detection, let us note that the goal of the tracking subsystem is to obtain an approximate location of a vehicle, lacking the precision needed to measure velocities or distances between vehicles. Now, once we have gathered the corresponding information about the tracked vehicles, the next step consists of aligning them with the foreground masks in the top-view of the scene, as they will allow us to compute accurate measures.

Figure 43. Vehicle trajectory trackers displayed as bounding boxes for a certain frame. Bounding boxes are simple and approximated representations of vehicles shapes and locations

In order to achieve mask-track alignment, we select the centre of each bounding box and transform them using the set of homography matrices H_t. By looking at which connected region the transformed point belongs to, we associate each tracked vehicle with a foreground element. By doing so, we pair the trackers with the foreground mask, so that each connected component in the foreground DA is aligned with one vehicle in the tracker dataset.

Fig. 44 shows the foreground DA from Fig. 42, but this time each connected component has been mapped to a vehicle trajectory tracker in the dataset, so each is highlighted in a different colour.

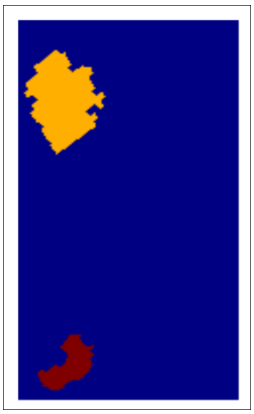


Figure 44. Foreground DA after mask-track alignment.

Hence, the output of this module is a set of indexed images DA_v, containing the foreground DA in which every tracked vehicle which lies on the region of interest is identified and aligned with one of the dynamic objects detected by the *background subtraction module*.

3.4.4 Event specification

This is perhaps the most complex module of the DVID subsystem. Being so, it has been further decomposed into three sub-modules, as it is shown in Fig. 45.

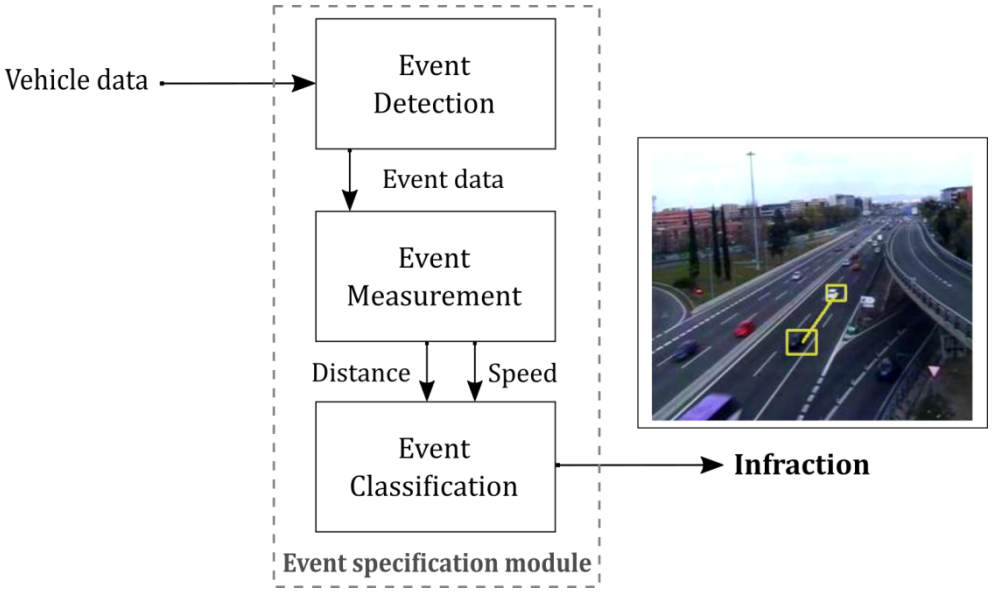


Figure 45. Event specification sub-modules. The event detection sub-module takes the information from the previous modules and establishes the condition for an event to happen. This event is handled by the event measurement sub-module, which takes the measures of speed and distance. These measurements are then evaluated by the event classification module to decide whether there has been an infraction or not.

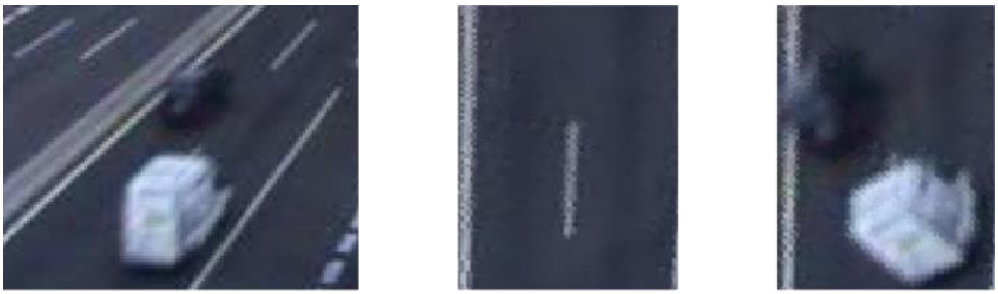
In order to detect certain traffic behaviours, we need to check if a set of conditions is met; this is done by *event detection sub-module*. Once an event has been identified, measures to characterize this event need to be computed by the *event measurement sub-module*. Finally, this event must be classified depending on those measures. The *event classification sub-module* is in charge of deciding whether the detected event is an infraction or not.

In this section, we explain the functionality of each of these sub-modules for the particular case of vehicle tailgating. However, it should be mentioned that this system could be almost directly used to detect any other traffic behaviour that can be characterized with measures taken in a normalised top-view environment.

3.4.4.1 *Event detection*

This sub-module specifies the condition for an event to happen. In the case under analysis (tailgating), we are looking for two contiguous vehicles that are close enough to the other. In other words, the first condition that we are looking for is that two or more vehicles are closely located in the same lane of the road. It is worth mentioning that the goal of this first sub-module is not to detect if the security distance is met or not, but to simply choose which cases are of interest to take later measurements. Hence, we aim to identify cases where vehicles are close enough to be studied more in depth.

In order to do so, we divide the DA in two lanes. We know that, due to the nature of the DA, any vehicle passing through must be either on the left lane or on the right lane. However, due to perspective, we must take into account that the vehicles on one lane might occupy areas of adjacent lanes. This effect becomes especially noticeable with tall vehicles (vans, buses or trucks), as it can be observed in Fig. 46.



(a) Original scene (b) Background Detection Area (c) Detection Area

Figure 46. Perspective limitation: Vehicle (white van) invading left lane.

The delivery van detected in Fig. 46.a is transformed according to the homography matrix of the nearest DL. In Fig. 46.c, we can observe how part of the white van spreads along the left lane in the top-view while, from direct observation, it is clear that the van is circulating on the right lane. Therefore, the camera perspective must be taken into consideration when specifying the limits of each lane. To this end, this sub-module specifies a default lane in which the vehicles are located unless proven otherwise. The default lane chosen depends on the camera perspective. If the camera perspective leaves the road oriented from bottom-left to top-right (Fig. 47.a) the default will be the left lane; whereas if the road is displayed from bottom-right to top-left (Fig. 47.b) the default will be the right lane.

(a) Left-lane dominant road (b) Right-lane dominant road

Figure 47. Road orientation effect on default lanes.

In order to locate a vehicle on the recessive (or non-default) lane, we specify a minimum area of the vehicle that has to lie within that lane. For the reader's better understanding of this process, we will subsequently explain the case of a left-lane dominant road (Fig. 47.a) so that, consequently, the right-lane dominance case can be extrapolated.

As it can be seen in Fig. 46, when a vehicle is on the left lane of the DA *(*dark private car*)* the vast majority of its area lies within the left lane. In the particular case of a tall vehicle, the rest of the area that does not lie within the lane would lie outside the DA (the median strip, in the example), so this would not be a problem. However, when a vehicle is on the right lane (white van), there is a high chance that a fair amount of its area will invade the left lane. From this, we can affirm that is less likely that left-lane vehicles invade the right lane than the opposite. As such, if a vehicle has a fair amount of its area lying within the right lane (even if it is not the majority), it is likely that the vehicle is circulating on the left lane. This means that the area of the vehicle located in the right lane could just be caused by the projection, as it happens in Fig. 46.c. Therefore, we establish the left lane as the default lane for every vehicle, and we specify a threshold on the minimum area that a vehicle has to have in right lane to be considered as circulating on that lane. This process is graphically displayed in Fig. 48.

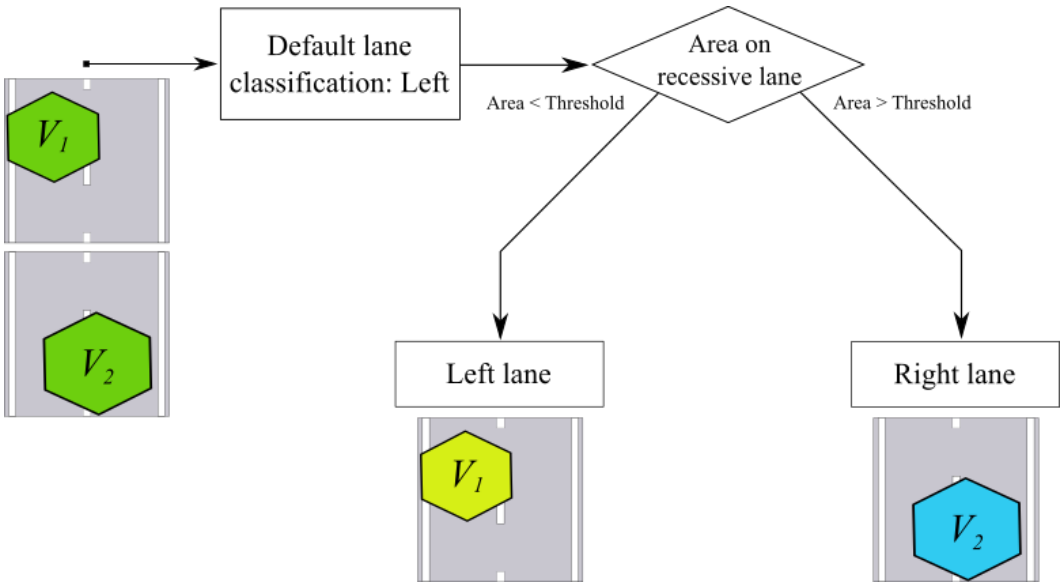


Figure 48. Lane classification flow chart.

After testing the process with different values for the threshold, a value equal to 30% of the total area of the vehicle was set. This means that, in order to classify a vehicle as circulating on the recessive lane (in this case: the right lane), at least 30% of its area must be located in the recessive lane or, in other words, in the right half of the image.

Also, in a similar way to lane discrimination, we must take into account that a vehicle that is neither on the right nor on the left lane might invade the DA due to perspective. For this, we simply specify a minimum area of the vehicle that must be inside a region of interest of the DA to be considered part of it.

Once the vehicles have been assigned to one lane, we must check if the condition for a potential event of interest is satisfied. As we explained in the beginning of this section, the condition is that, within a DA, there are two or more vehicles in the same lane. As the vehicles in the DA have already been associated to one lane, it is just a matter of counting. If the condition is satisfied, we proceed to take measures to check whether the security distance is met or not.

3.4.4.2 *Event measurement*

Once an event of interest is detected, this sub-module measures the involved magnitudes to evaluate if the security distance is met or not. For the particular case of tailgating, we have two measures of interest, namely: a) *the distance between both vehicles*, and b) *the speed at which the rear vehicle is moving* (it only takes into account the speed of the vehicle behind, as it is the one causing the violation). It is important to note that, as the domain in which the event is detected has been normalized, we can neglect the horizontal coordinate and simply perform a subtraction between the vertical coordinates of each point.

In order to calculate the aforementioned measurements, the system first establishes some reference points. In the case of the distance, the calculation is based on the two points belonging to adjacent vehicles whose limits in the vertical component are closer; this are the top of the vehicle located at the bottom of the image (p_{bv}; marked green in Fig. 49) and the bottom of the vehicle located at the top of the image (p_{tv}; marked red in Fig. 49).

$$d_v = p_{bv} - p_{tv} \tag{17}$$

where d_v is the distance between vehicles. Note that the value of the vertical coordinates increase from top to bottom.

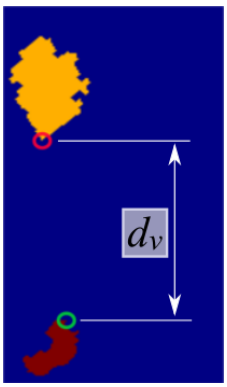


Figure 49. Points selected to measure distance between vehicles. The red dot represents p_{tv}; the green dot represents p_{bv}.

It is worth mentioning that the distance between vehicles at a particular instant does not require information from previous frames, so it is only measured when an event is detected. In contrast, the speed of the vehicles is obtained for every vehicle passing through a DA, even if the vehicle does not need any further processing (for example, when there is no other vehicle in the same lane). This is done due to the need of at least two frames to compute the speed. Hence, the system requires measuring the speed of every vehicle passing through a DA.

The method to do so is very straightforward: we define f_o as the frame in which the vehicle enters a DA; and p_o is then the position of the vehicle in said frame. Then for any given frame f_i, the speed of the vehicle could be calculated as:

$$v_{pxl} = \frac{p_i - p_o}{f_i - f_o} \tag{18}$$

In order to obtain a smooth estimation of the speed, we consider it to be constant along the DA, and hence we can assume that, for the practical calculations, f_i is going to be the last frame in which the vehicle is present in the DA.

It is important to note that this speed is estimated in pixels per frame; therefore, in order to obtain a real magnitude, we need to know the frame rate at which the video was taken, which is fixed, and the correspondence R_{pxl} between pixels in the normalised top-view and meters in the real scenario. The latter is established prior to obtaining the homography matrix. Then, the speed is simply:

$$v = \frac{p_f - p_o}{f_f - f_o} \cdot \frac{R_f}{R_{pxl}} \cdot 3.6 \tag{19}$$

where R_f represents the frame rate in frames per second; R_{pxl} represents the ratio between pixels and meters (expressed in pxl/m); v represents the speed in km/h.

These measurements are the output of this sub-module, and will be used to classify the event to detect possible infractions.

3.4.4.3 *Event classification*

This sub-module takes as input the values that define the event. For the tailgating case, these are the speed and the distance between vehicles. Based on this input, it estimates the danger of the situation. Depending on this danger factor, the system decides if an infraction exists.

We have defined a factor that models the danger of a crash between vehicles: *the time of impact* t_i:

$$t_i = \frac{d_v}{v_r} \tag{20}$$

where the parameter v_r is defined as the speed of the rear vehicle and d_v is the distance between adjacent vehicles. The time of impact represents the time needed by the rear vehicle to impact the next one in case of a sudden stop. Assuming that both vehicles need the same time to stop once the brake pedal is pressed, the time of impact is then the time that the driver in of the rear vehicle has to react to a sudden break of the previous vehicle. If this reaction time is below a threshold, the system will mark the event as a tailgating infraction.

3.5 System output

The final system output is a set of pairs of vehicles driving in the same lane with their corresponding times of impact t_i. By setting up an absolute threshold over t_i, one could determine whether or not tailgating is taking place. Unfortunately, due to the lack of official regulation, this threshold is not specified and only recommendations can be found in the literature.

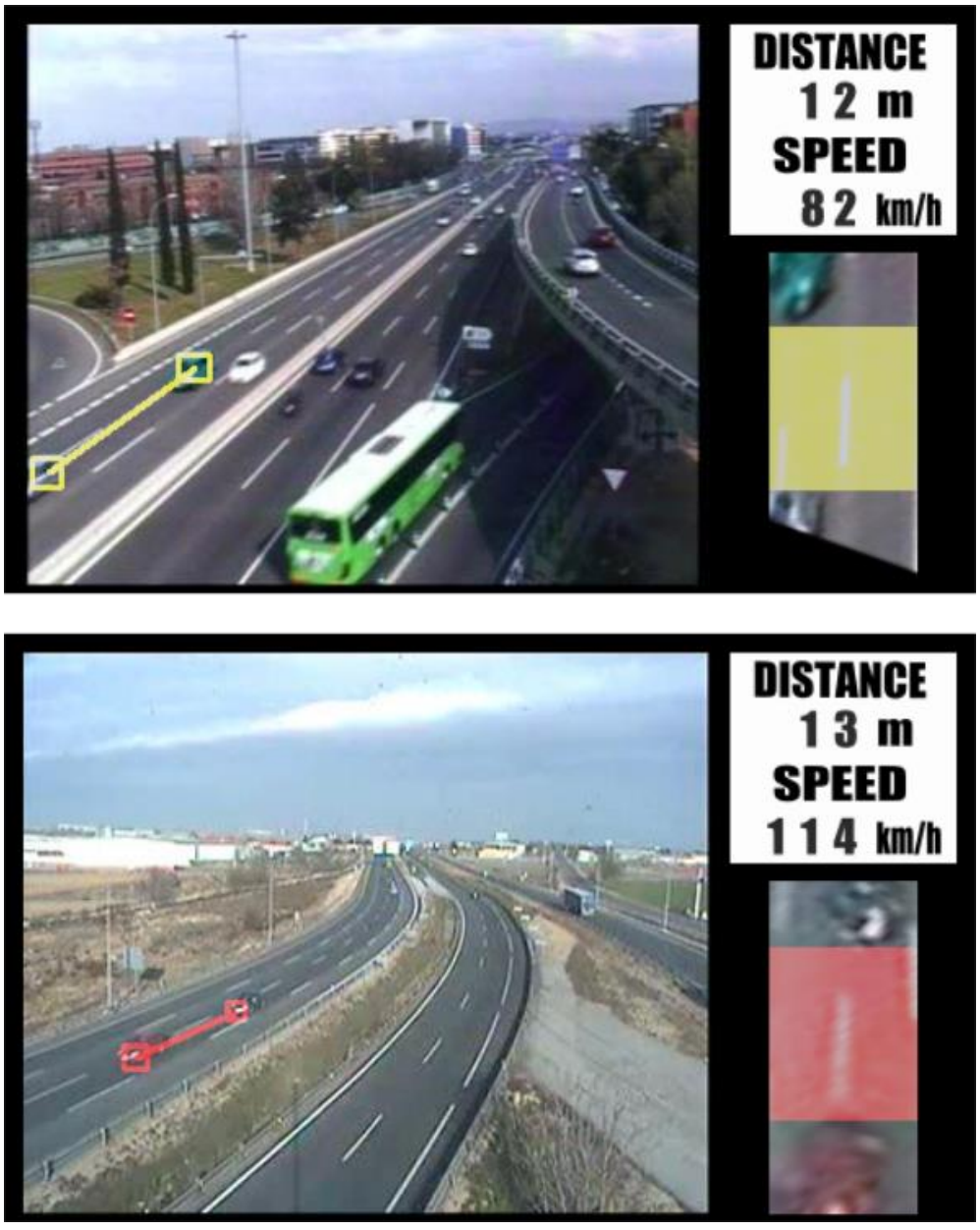


Figure 50. Output visualization examples; yellow marks moderate infraction, red marks severe infraction.

In Fig. 50 an example of the final output of the system is shown, identifying involved vehicles and measuring speed and distances. Of course, the actual output is the data of the infraction (vehicles involved, frame range in which it takes place, distance between vehicles, speed of the rear vehicle, etc.), but Fig. 50 displays a possible visualization of one particular infraction at a certain time.

3.6 An additional discussion about design alternatives

The design and implementation of this project was not a straightforward process. Several possibilities were considered during each stage of the development; decisions were made and alternatives were discarded for various reasons. In this section, we will present the reader with the most significant and relevant design alternatives that were considered during the development of the project. In each case, we will discuss the differences with the proposed design and justify the choice.

3.6.1 Unique transformation matrix

According to the current design, several transformation matrices are estimated for each scene, each one centred in a DL and associated to a DA. However, one cannot help but wonder if it would have been simpler to establish a unique transformation matrix for the whole image affecting to all the DAs. Indeed, this option was the first considered in the design of the system. However, there are several limitations that prevented this alternative.

The main reason is the noise of the image. If we take one DL as the centre of the transformation, we can use the homography matrix obtained to transform the whole image. This case is shown in Fig. 51. As it can be observed in the image, the surroundings of the chosen element have enough resolution and equivalent proportions to an ideal normalised environment. However, the farther we go from the analysed DL, the noisier the image gets. Therefore, the measures taken for the vehicles passing far from the DL would be dramatically less accurate than the ones taken for the ones near it.

Figure 51. Unique transformation using a single element.

Then, we can think of a different option. We could take into account more than one DL to build a general transformation matrix. However, there is no way of assuring that we are detecting every single DL in the scene, so we cannot accurately state the spatial relation between detected DLs, and even if we could, the out coming transformation matrix would provide a map that is accurate for the areas near detected DLs and dramatically loses precision as we get away from those areas. Therefore, in order to choose quality of detection over quantity, it was decided that every DL would generate a separate region of interest in which measures could be taken.

The most important downside to this decision is that we cannot compare vehicles that are located in different regions of interest. However, as we are measuring tailgating, we can

consider that two vehicles do not represent a threat to road safety if they are located sufficiently far from each other so that they appear in different regions.

3.6.2 Pre-transformation mask-track alignment

Mask-track alignment allows us to establish a relationship between the dynamic objects we find in the scene and the vehicle information stored in the vehicle trajectory trackers. Before recurring to the current method of mask-track alignment, a different alternative was considered.

This alternative method consisted in selecting the points needed to take the measures in the original context, and transform just those points to save the computational cost of transforming the whole image. This also allows us to perform the mask-track alignment before applying the transformation, thus resulting in a straightforward process of location matching.

To define the vehicle, three points of interest are selected:

- Top point: represents the part of the vehicle with lower vertical coordinate. It will mark the distance with the vehicles in upper parts of the scene.
- Bottom point: represents the part of the vehicle with higher vertical coordinate. It will mark the distance with the vehicles below in the scene.
- Lane point: identifies the part of the vehicle that is closer to the road: its base. It will be used to determine the lane in which the vehicle is circulating.

The location of these three points depends on the camera perspective, and they are always chosen from among the extrema of the object (see regionprops[3]). Fig. 52 shows the selection of these points. As it can be observed, the bottom point will be obtained from the lowest point of the object (if there is more than one bottom point, we distinguish between bottom-left and bottom-right depending on the camera perspective). For the top point, the same reasoning is followed with the highest point of the object. Finally, in order to find the lane point, we determine two points: bottom-left and left-bottom in right-lane dominant roads; bottom-right and right-bottom in left-lane dominant roads. The lane point is the middle point of the segment conforming the two mentioned points.

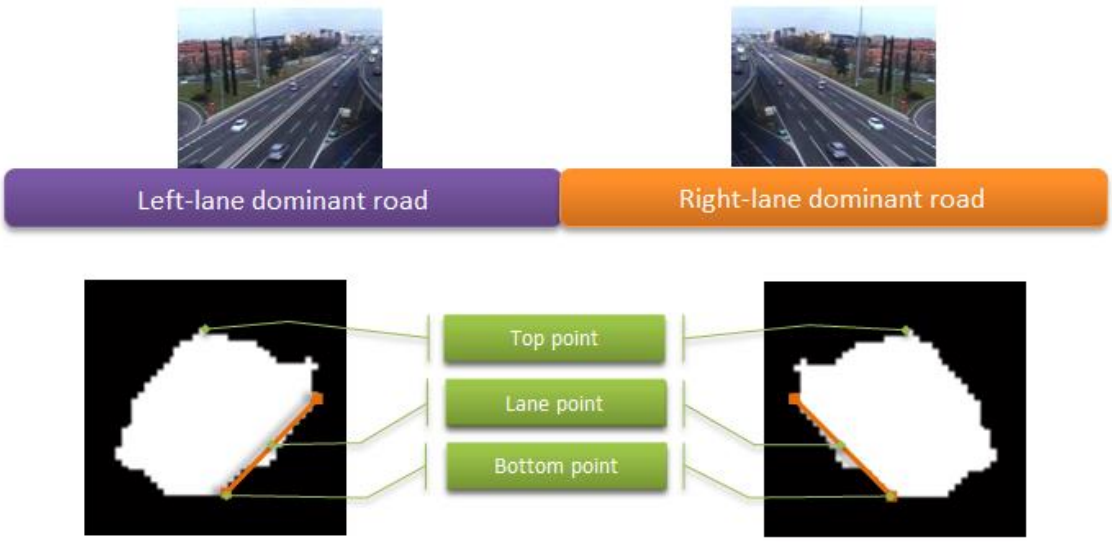


Figure 52. Representation of the points of interest.

[3] MATLAB function *regionprops*: http://es.mathworks.com/help/images/ref/regionprops.html

These points are then transformed to follow a similar procedure to the one in the *event specification module* explained in Section 3.4.4. In order to determine the lane in which the analysed vehicle is circulating, the transformed lane point horizontal coordinate is checked. As the lane point does not represent the centre of gravity of the vehicle, but rather the visible part of the vehicle that is closer to the road, the threshold for selecting the lane will not be the half of the DA, but rather a shifted version of it (the direction of the shift will depend on the camera perspective). Fig. 53 shows the selected limits for each lane in a left-lane dominant road. Additionally, it should be stated that there is also an exterior limit for both limits, in order to prevent vehicles present in lanes outside the DA to interfere with it, disturbing the analysis. It is worth mentioning that the right-hand exterior limit is located outside the limit of the road itself, as the noise in the foreground mask, together with the noise added by the transformation, can sometimes cause this point to be shifted beyond the limit of the lane to which they actually belong.

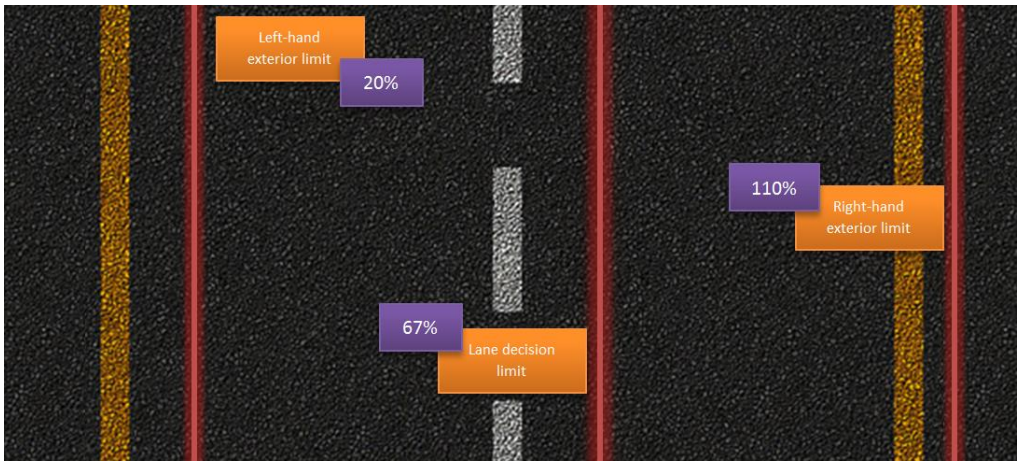


Figure 53. Lane decision limits.

After locating two or more vehicles in the same lane, the distance between the adjacent ones is measured by subtracting the vertical coordinates of the transformed top and bottom points of the vehicles.

This method has several disadvantages. The first disadvantage is that it is much harder and imprecise to track the speed of the vehicles, thus leaving us with a tailgating detector based solely on the distance between the vehicles. Another important drawback is that the accuracy of transforming the points of interest is much lower than if we transform the whole image and look for the points in the top-view environment. As a result, it was decided that a computationally heavier algorithm was the lesser of two evils.

4 Experiments and results

In this chapter, we describe the experiments carried out and the results obtained, which are used to evaluate the performance of the method described in chapter 3. First, we will present the dataset and the experimental setup that were used to test the implemented algorithm. A detailed description of the experiments conducted will be then presented. To conclude, we will also comment on the results obtained from these experiments.

4.1 Dataset and experimental setup

The dataset used for the project's assessment was provided by the Spanish Traffic Management Administration (Dirección General de Tráfico – DGT) and it consists of different videos recorded by the CCTV system of the aforementioned institution.

As we will see throughout this chapter, some of the available material from this dataset does not satisfy the requirements specified in Section 3.1.2. For instance, the resolution of the videos is rather poor (352x288 pixels) and the recording frame rate is also not very high (25 fps). This is because this dataset was not intended for this kind of processing. In Chapter 6 we will suggest a few possible modifications for the dataset conditions in order to obtain better results in future implementations.

The videos show real scenarios (camera locations) with varying traffic situations. The current dataset consists of 7 camera locations. For each location, we have several one hour videos covering the period between 8 a.m. and 17 p.m. During this period of time, as the cameras are motorized, the scene view changes several times a day due to rotations and zooms made by the camera operators. As a result, we have obtained as much as 80 different background images in which the geometric models can be calculated. Some illustrative examples of this background models are shown in Fig. 54.

Figure 54. Background images from different camera locations.

In an attempt to evaluate the performance of the system, several experiments have been conducted, yielding key information to achieve a better understanding of the output obtained. In the following sections, we will analyse those experiments and interpret their results.

4.2 Experiments on road lane classification

As it was explained in Section 3.3.2, this system has two different methods that can be used to classify the elements of the road into the selected categories (DL and BL). In order to determine the performance of the considered methods, a comparative study was performed.

	Automatic	Semi-automatic
Total	79	142
Excellent	45	109
Acceptable	11	30
Defective	3	3
Incorrect	20	0

Table 4. Total number of homographies calculated. Human error absence is assumed for semi-automatic method.

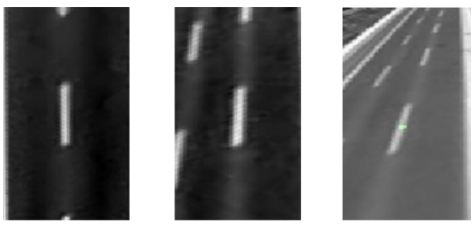
Table 4 shows a relation between the number of DAs found in the whole sample space for each method. From a strictly objective point of view, we can confirm at first glance that the performance of the semi-automatic method is better, as the total number of homographies found with this method is almost twice the number found with the other. A subjective study was also conducted to evaluate the quality of the obtained transformations, yielding positive results for both methods but, again, better for the semi-automatic method. In this subjective study, four labels were created to define the quality of the homography performed:

Excellent: A transformation is considered excellent if the resulting transformed image is very similar to the ideal output, with both lanes clearly separated and visible (Fig. 55.a).

Acceptable: A transformation is labelled as acceptable if the matrix is usable and measures can be taken, but it presents some artifacts or defects, like a soft rotation or slightly incorrect shear (Fig. 55.b).

Defective: A transformation is considered defective if it cannot be used or if the measures taken from the transformed domain are expected to be totally incorrect (Fig. 55.c).

Incorrect: A transformation is labelled as incorrect when it is not performed on a desired element. This can happen with DLs or BLs that are not completely contained in the scene, thus resulting in a wrong estimation of the measures.



(a) (b) (c)

Figure 55. Qualities of homography (Excellent, acceptable and defective).

As we can see, the automatic method does not always detect correct homographies, as it does not distinguish actual DL from incomplete DL or BL. The semi-automatic method deflects this issue with user involvement in the process. The automation of this kind of discrimination is reserved for future lines of work.

If we analyse the performance of each method in the 7 different camera locations, we can get a deeper insight into the behaviour of the system, understanding where it works better and why. It is important to note that each location is filmed by a single camera operated by a human. Throughout the day, it changes its perspective several times; however, there is not a fixed pattern for these camera movements. Fig. 56 shows the performance for both methods in the different camera location, which will be explained below.

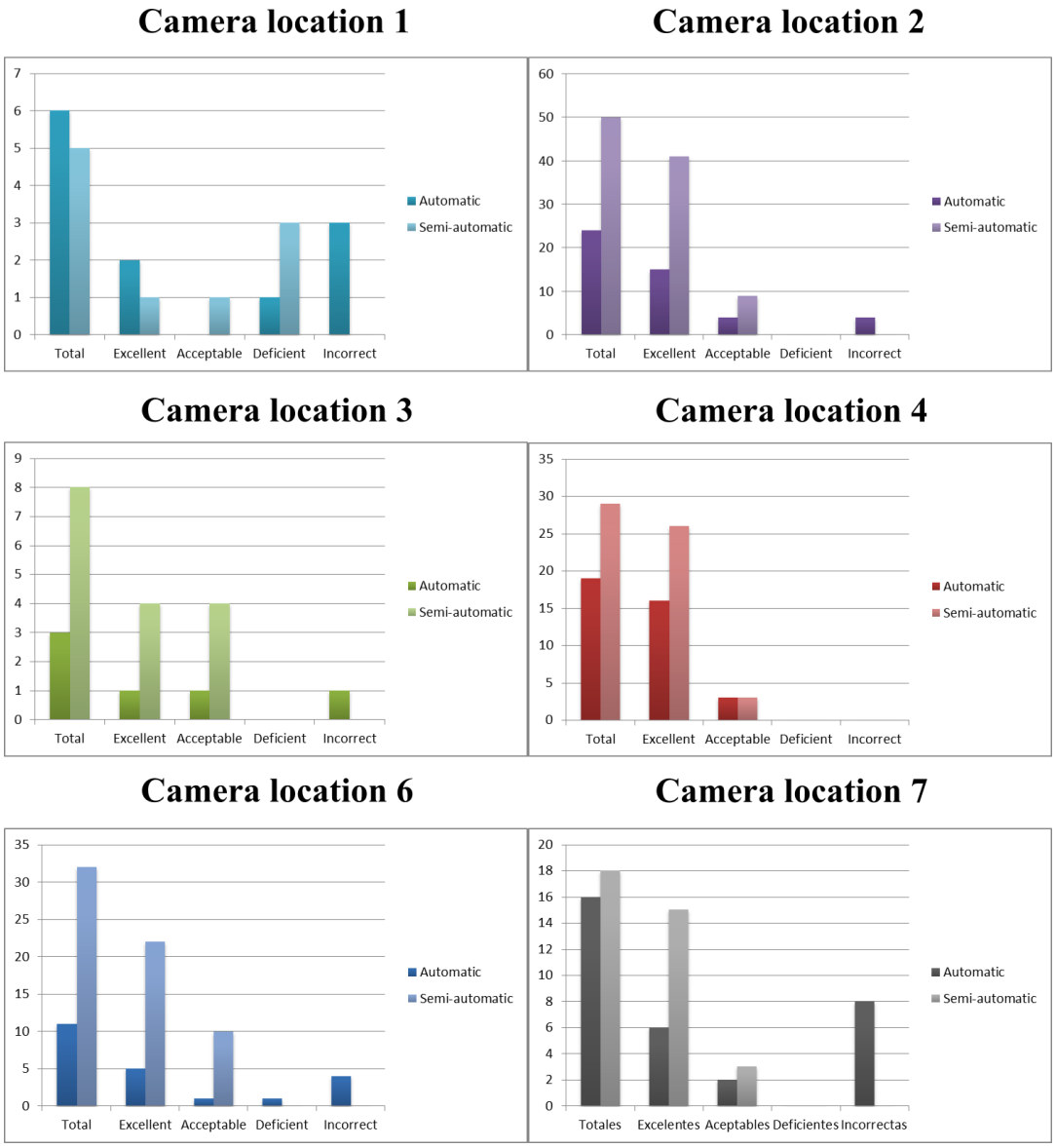


Figure 56. Performance of automatic and semi-automatic methods is displayed for the different camera locations. Note that location 5 is missing, as it does not produce any usable geometric models.

4.2.1 Camera location 1

This camera location presents three main scenes throughout the day. The first shows two sets of lanes in both directions (Fig. 57.a). We can see at first sight that this image is going to present some problems (especially for the automatic method), as there is a tree and a lamppost blocking the sight of the road. The second scene shows a similar scenario, but no elements are blocking the road (Fig. 57.b). This scene presents multi-lane scenarios with some tricky situations, especially taking into account that the lighting is not optimal at any point of the video for this case, but it works better than the first case. The last case is the same as the second, but zoomed out (Fig. 57.c). As the resolution of the image is not very good, it is easy to imagine that the farther the road is, the worse detection will be. In the example in Fig. 57.c, neither the automatic nor the semi-automatic method was able to find usable homographies.

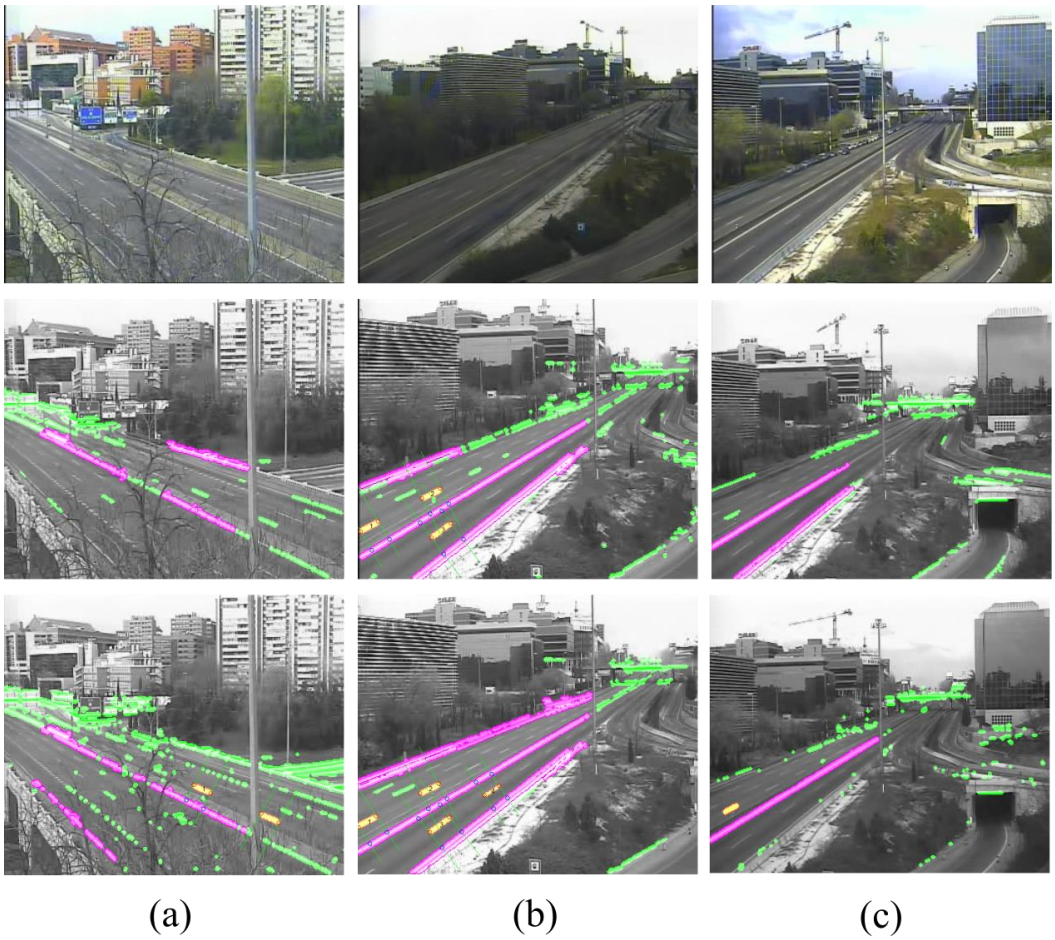


Figure 57. Performance of automatic method (second row) and semi-automatic method (third row) is shown for different scenes of Camera location 1. Marked in yellow with a small number the DLs detected can be observed.

4.2.2 Camera location 2

In this location there are only two main scenes. The first one is completely useless, as potentially useful roads are strongly occluded by a bridge (Fig. 58.a). The second one, however, is very useful, as it contains a very clear view of a busy road (Fig. 58.b). Fortunately, this is the scene that remains visible for most of the day, and the lighting is sufficient for the system to

extract reliable measures on several DLs. In fact, this is the camera location that provides more DAs and, consequently, more infraction alerts.

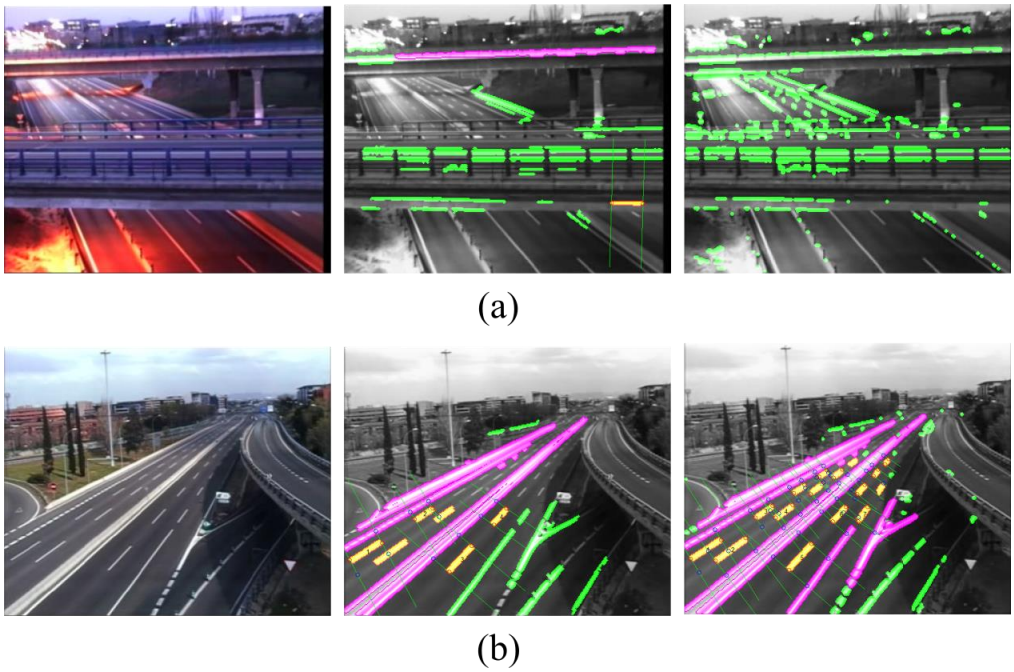


Figure 58. Performance of automatic method (second column) and semi-automatic method (third column) is shown for different scenes of camera location 2. Marked in yellow with a small number the DLs detected can be observed.

4.2.3 Camera location 3

This camera location contains three different views. Two of those views have presented a problem that has been impossible to overcome in this project. The camera is oriented in a very small angle with respect to the road. This perspective causes (as it can be appreciated in Fig. 59.a and Fig. 59.b) the DLs to be very short in length and, therefore, impossible to detect, as our detector is based on the eccentricity of the objects. The third view is remarkably clear and provides several useful DAs. Unfortunately, unlike the case of camera location 2, this scene is the least common throughout the day, so this camera location does not provide us with sufficient information to extract valid infraction alerts.

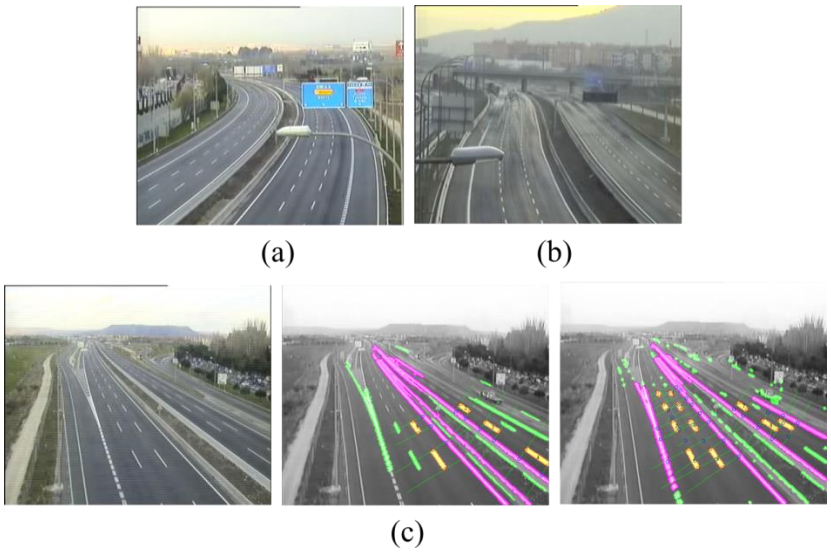


Figure 59. Unusable views due to perspective (a and b) and performance of both methods for camera location 3 (c).

4.2.4 Camera location 4

This camera location is formed by a single view (Fig. 60). The scene is clear, although the traffic is not very dense, so even though the geometric models are great, this video will produce few infraction alarms. Again, we can clearly appreciate that the semi-automatic method works better.

Figure 60. Performance of automatic and semi-automatic method in camera location 4.

4.2.5 Camera location 5

This location is also formed by a single view. However, as it can be observed in Fig. 61, the road is too far away from the camera and the brightness of the scene does not produce sufficient contrast for the edge detector to work properly, as there are great amounts of noise from the rest of the image.

Figure 61. Main view for camera location 5.

4.2.6 Camera location 6

This is also a single-view scenario (Fig. 62). In this case, the view shows a part of the road that can be analysed, as the road lines are clear enough to distinguish. The lanes on the left, however, are too far from the camera, so the noise impedes further processing.

Figure 62. Performance of the automatic (middle) and the semi-automatic (right) method for camera location 6.

4.2.7 Camera location 7

This last location is also formed by a single view. Even though the illumination does not allow us to get great geometric models throughout most of the day, there are certain times of the day that provide us a clear view of the road, like the one displayed in Fig. 63. With the current system, we are only able to use the geometric models corresponding to a one-hour period within that hour, even if the characteristics of the image at another given time are essentially the same. However, future lines of work may include a method to use any models throughout the rest of the day, improving the detection rates significantly (see Section 6.2.4).

Figure 63. Performance of the automatic (middle) and the semi-automatic (right) method for camera location 7.

In Fig. 63 we can also appreciate a case of incorrect detection, with the lowest DL detected with the automatic method. This DL is not completely contained in the image, so we do not know its exact length, and further processing would result in inaccurate measures. The semi-automatic method is able to tackle this matter through human interaction.

4.2.8 Conclusions about the recommended camera view

Now that a detailed analysis on the camera locations has been performed, we can begin to discuss which would be the ideal camera settings for the system described in this manuscript to work effectively and efficiently.

Regarding the altitude of the camera, the case is simple: the higher it is, the closer we are to our ideal top-view and the less occlusion we will suffer. Therefore, a higher altitude is preferred

for the system to work properly. Nonetheless, there might be some cases in which a significantly high altitude for the camera is not suitable, as there might not be a place to attach it, or the view of the road from such place might be non-sufficient.

The analysis conducted during the experimental phase of this project yielded better results for camera locations in which perspective allowed the discontinuous lines to provide a long enough projection in the camera image plane. In our database, these locations had the road oriented diagonally with respect to the camera image plane. Meanwhile, those locations that offered views in which the discontinuous lines were perpendicular to the image plane provided worse or even null results. From this, we deduce that, if the altitude of the camera is not sufficiently high, there is a need to find long projections of the discontinuous lines by filming from the side. However, a perspective in which the road is almost parallel to the camera plane could lead to unmanageable cases of occlusion.

As a result, the conclusion drawn from the analysis is that, if high altitude positions for the camera are not available, the best decision for the system to work is to film the road so that discontinuous lines are seen diagonally from the camera plane, at around 45º, as it happens in camera location 2.

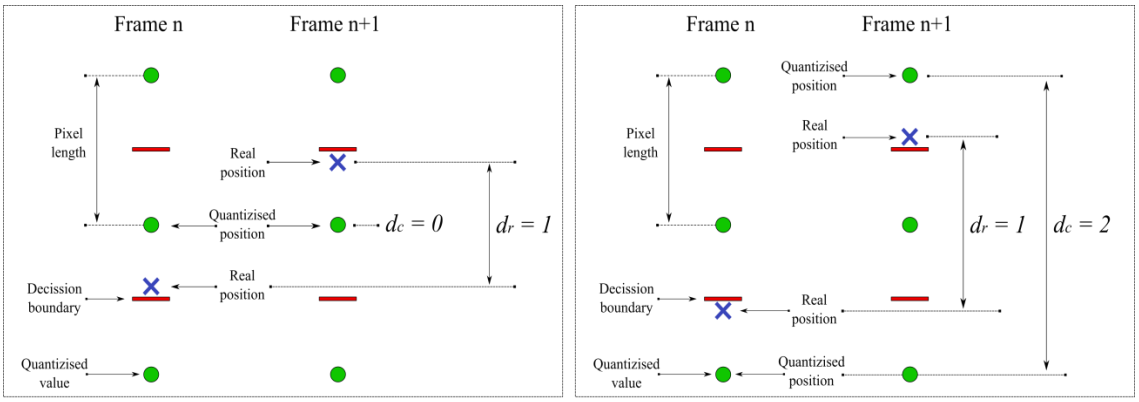
4.3 Theoretical study on the precision of the measurements

It is important to notice that the system has external limitations that cannot be avoided by any image processing techniques. The video resolution (spatial and temporal) strongly affects the system performance. As any digital system, this one has errors due to quantization.

In order to understand how much the system is affected by this factor, we need to study the uncertainty around the measures we are taking. For this, a theoretical study on this uncertainty was performed. As it can be observed in Section 3.1.3, there are different measures for the DLs depending on the type of road. In order to avoid unnecessary complications in the experiments, for this study it was assumed that all the provided camera location belonged to the case represented in Fig. 16.b (see Section 3.1.3): roads with the maximum speed range of 60 to 100 km/h. Therefore, the theoretical length of a DL for this study is considered to be:

$$L_m = 3.5\ m \tag{21}$$

We need to know the minimum distance that we can perceive between frames. Fig. 64 shows the advance of a point from one frame to the next. As it can be observed, the worst case scenario between frames gives us an error of 1 pixel. In Fig. 64.a, the distance is calculated as $d_c = 0$, when it is actually $d_r = 1$; whereas in Fig. 64.b it is calculated as $d_c = 2$, when it is actually $d_r = 1$. Note that d_r stands for real distance, whereas d_c stands for calculated distance.



(a) Worst case by proximity (b) Worst case by remoteness

Figure 64. Worst cases for the spatial error. The green circles represent the position of each vertical edge of the pixels, and the red markings represent the decision boundary that determines to which pixel the measured position (marked as a blue cross) gets quantizised.

Therefore, we can easily deduce that the spatial error is going to be:

$$e_{pxl} = 1 \left[\frac{pxl}{frame}\right] \tag{22}$$

However, this error will represent a different measure in meters depending on the DL that we are analysing. Expectedly, the farther the DL is from the camera, the bigger the error. If we consider the transformation to keep the resolution of each line fixed; in other words, if we consider the length of the transformed DL to be the same as the length of the original DL, we can express the error in terms of this length in pixels (L_{pxl}):

$$e_x = \frac{L_m}{L_{pxl}} \cdot e_{pxl} \left[\frac{m}{frame}\right] \tag{23}$$

This represents the spatial error from frame to frame. If we want to express the error in the calculation of the speed, we have to factor this with the frame rate R_t, being:

$$e_v = e_x \cdot R_t = \frac{L_m}{L_{pxl}} \cdot e_{pxl} \cdot R_t \left[\frac{m}{s}\right] \tag{24}$$

For the sake of simplicity, we have decided to express them in km/h, so we can finally express the speed error as:

$$e_v = 3.6 \cdot \frac{L_m}{L_{pxl}} \cdot e_{pxl} \cdot R_t \left[\frac{km}{h}\right] \tag{25}$$

Let us consider the image in Fig. 65. Then, we can calculate the expected error for the three measured DLs:

$$e_{x1} = 3.6 \cdot \frac{3.5}{27} \cdot 1 \cdot 25 = 11.67 \left[\frac{km}{h}\right] \tag{26}$$

$$e_{x2} = 3.6 \cdot \frac{3.5}{18} \cdot 1 \cdot 25 = 17.50 \left[\frac{km}{h}\right] \tag{27}$$

$$e_{x3} = 3.6 \cdot \frac{3.5}{13} \cdot 1 \cdot 25 = 24.23 \left[\frac{km}{h}\right] \tag{28}$$

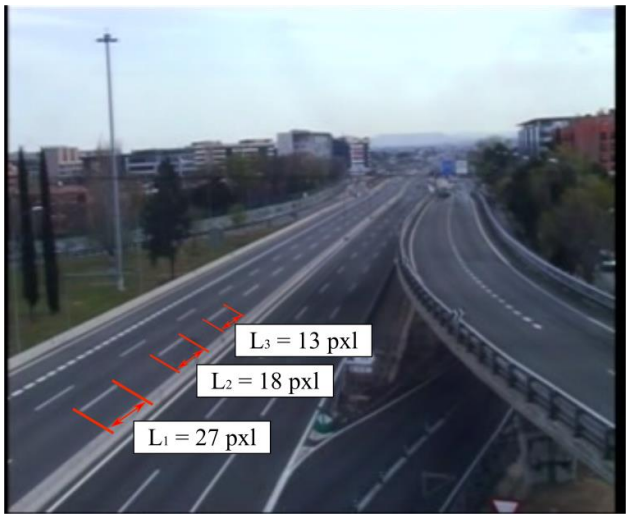


Figure 65. DL lengths for a specific camera location.

As it can be observed, the error is high even for the best cases. Moreover, this error will be increased by the uncertainty caused by the detection of points, transformations and further processing performed into the images. Therefore, this would be unacceptable for a real system, especially taking into account that this particular error could be greatly reduced easily.

For instance, let us consider the new image resolution to be 1080p (1920x1080 pxl) while the scene remains to be the same. Taking into account that our current resolution is 352x288 pxl, and taking the most restrictive ratio, the spatial resolution would improve according to a factor f_{Rx}:

$$f_{Rx} = \frac{1080}{288} = 3.75 \tag{29}$$

Additionally, let the new frame rate be 60 fps, then we can establish a temporal resolution factor of:

$$f_{Rt} = \frac{60}{25} = 2.4 \tag{30}$$

Then, the total resolution factor to be considered would simply be:

$$f_R = f_{Rx} \cdot f_{Rt} = 9 \tag{31}$$

With this resolution, we can recalculate the error as:

$$e'_v = \frac{e_v}{f_R} \left[\frac{km}{h}\right] \tag{32}$$

Thus, the new values for the errors in the scene represented in Fig. 65 would be:

$$e_{x1} = \frac{11.67}{9} = 1.29 \left[\frac{km}{h}\right] \tag{33}$$

$$e_{x2} = \frac{17.50}{9} = 1.94 \left[\frac{km}{h}\right] \tag{34}$$

$$e_{x3} = \frac{24.23}{9} = 2.69 \left[\frac{km}{h}\right] \tag{35}$$

As we can see, the possible error would reduce drastically if the resolution of the cameras were better. As we will see in Chapter 6, future lines of work include the generation of a dataset with better resolution parameters.

4.4 Statistical study on the quality of the measurements

Up to now, we have established the theoretical limitations of the system. Nevertheless, we have not yet found an experimental method to evaluate the performance of our approach. The goal of the system proposed in this project is the automatic detection of infractions related to tailgating (see Section 3.5). Unfortunately, we currently lack an objective protocol to evaluate this output, as there are no known datasets concerning tailgating detection. Additionally, we lack ground truth information about vehicle velocities and distances. Even though we have thought of future implementations which may include methods to verify the obtained information in different ways (see Section 6.2), we still need a method to evaluate the performance of the system in some sense.

Direct observation of the results over various traffic sequences has allowed us to check that, when the majority of the requirements specified in Section 3.1.2 are satisfied, the system successfully detects those cases where tailgating is very clear. Nevertheless, this evaluation is rather subjective, and we have therefore performed a study about the precision and robustness of those measures that support the decision of the tailgating detector. In particular, we have chosen the speed of the vehicles as the variable being quantitatively evaluated, as it provides a general idea about the quality of both spatial and temporal measures. As we have seen in the previous section, the uncertainty of the measures varies depending on the length of the DL. For this reason, the error will vary from one camera location to another. For this study, we have calculated a general error on the speed of each vehicle (do not get confused with the theoretical error calculated in Section 4.3). The method for computing this error is rather simple and again assumes that vehicles move at a constant speed during the set of frames in which it is located within the DA.

First, the **average speed** of a vehicle in an area of interest is estimated using the method implemented in our proposal (see Section 3.4.4) according to (22). As this value $\bar{v}$ is in pixels, but it can be non-exact, we determine:

$$\overline{v_{max}} = \lceil \bar{v} \rceil \tag{36}$$

$$\overline{v_{min}} = \lfloor \bar{v} \rfloor \tag{37}$$

where the operator $\lceil - \rceil$ represents the next integer number (superior) and the operator $\lfloor - \rfloor$ represents the previous integer number (inferior).

Then, instant (frame by frame) values of the speed are estimated:

$$v_i = \frac{p_i - p_{i-1}}{f_i - f_{i-1}} = p_i - p_{i-1} \tag{38}$$

where f_i represents the current frame and p_i represents the position of the vehicle at that frame. Note that the disappearance of the denominator is due to the assumption that we analyse just one frame at a time, so it yields that $f_i - f_{i-1} = 1$.

We then calculate an error for this instant speed based on the value of the general speed $\bar{v}$. In order to obtain the error related to our empirical measures as accurately as possible, we allow the minimum value of the difference between the instant and the integer limits of the average speeds. This is:

$$e_i = min(|v_i - \overline{v_{max}}|, |v_i - \overline{v_{min}}|) \tag{39}$$

Hence e_i stands for deviation of the instant speed with respect to the average speed, and should be null for all frames in the case were a vehicle is moving with constant speed. Finally, the general empirical error of the speed e_v for one particular vehicle is calculated as a temporal average error. From (39), it yields:

$$e_v = \frac{1}{n} \cdot \sum_{i=1}^{n} min(|v_i - \overline{v_{max}}|, |v_i - \overline{v_{min}}|) \ \left[\frac{pxl}{frame}\right] \quad (40)$$

Of course, this error is expressed in $pxl/frame$. As we explained in the previous section, the value for the error in km/h would depend on the line analysed, but in this study we will choose a general pixel-to-meters ratio, in order to facilitate the understanding of the process. The chosen ratio for the study is:

$$r_x = 15 \ \left[\frac{pxl}{m}\right] \quad (41)$$

This means that each meter is represented by 15 pixels in the transformed top-view of each DA. Then, we can express the error in km/h as:

$$e_v = \frac{3.6 \cdot R_t}{r_x} \cdot \ \frac{1}{n} \cdot \sum_{i=1}^{n} min(|v_i - \overline{v_{max}}|, |v_i - \overline{v_{min}}|) \ \left[\frac{km}{h}\right] \quad (42)$$

This error has been calculated for every vehicle passing through the scene for each camera location available. The results of this statistical study are shown in Table. 5. Our assumption of constant speed will theoretically generate null deviations from the averages, so that the non-zero values in the errors are mainly due to limitations on the precision of our method.

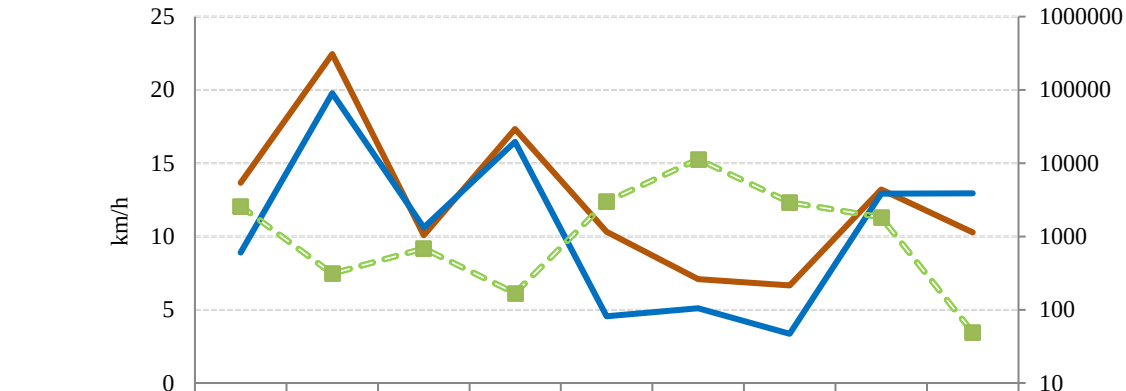
Camera Location	Error Average (km/h)	Error Deviation (km/h)	Samples (Vehicles)
1	9,28	11,14	2975
2	8,50	5,26	17146
3	----	----	0
4	11,51	12,37	2607
5	----	----	0
6	27,95	20,13	107
7	21,25	15,64	70

Table 5. Statistical results for the speed error in the different camera location.

As we can see from Table 5, there are significant differences in detection from one video to another. This, of course, must be taken into account for future implementations, as it may help deciding the characteristics of new infrastructure (camera locations, perspective, etc.).

Along with the analysis under the location point of view, we also performed a study focusing on the Time of Day. The results of this study are shown in Fig. 66. We have determined that the best moment for detection is between 12:00h and 15:00h. Incidentally, the lighting conditions during these hours allow a very clean view of the road, which has led us to believe that these are the best conditions for our system to work on. During these hours, our system reaches its best detection rate, with a peak of 11341 samples in the range from 13:00 h to 14:00 h. Although the

absolute number of detections may strongly depend on the amount of traffic, we can see in the figure that the speed error also reaches its minimum values in this interval, with average values around $7km/h$, along with a small deviation ($3.36\,km/h$ at 14:00 h). This later observation supports our previous conclusion that this period of times provides the best illumination conditions for our application.



	08:00 h	09:00 h	10:00h	11:00 h	12:00 h	13:00 h	14:00 h	15:00 h	16:00 h
Error Avg	13,66	22,45	10,07	17,34	10,32	7,09	6,66	13,21	10,28
Error Dev	8,91	19,77	10,60	16,45	4,55	5,10	3,36	12,93	12,95
Samples	2592	311	690	168	2985	11341	2939	1830	49

Figure 66. Error analysis by hours showing the average error (red), the error deviation (blue) and the number of samples (green) each hour. The right vertical axis marks the number of samples, while the left vertical axis marks the speed in km/h. Please note that the character ‘,’ is used to separate the integer from the decimal part.

Therefore, we can observe that in the early hours of the morning the system will perform worse than after noon. Of course, as this is presumably mainly due to the lighting and the weather conditions, the trend could change for videos recorded during a different periods of the year.

In general, the results of this experiment are quite promising, as we are able to estimate the speed of the vehicles in varied scenarios with a relatively small margin of error (given the resources available). However, from our point of view, the system may not be precise enough to monitor tailgating in the case of having as a goal imposing fines and penalties. In Chapter 6 we will discuss potential improvements of the system performance that might help us to reach this objective.

5 Planning and budget

This project is included in a scholarship of the Signal Theory and Communications department at the Carlos III University. Hence, a specific schedule was developed and will be described in the chapter. We will also present a brief description concerning the relevant instruments, such as tools or facilities, which supported the development of the project. After that, a section will be devoted to the information regarding the budget of the project.

5.1 Project schedule

This project is a continuation of the research work performed in a previous research project of the Multimedia Processing Group of the Signal Theory and Communications Department (the interested reader is referred to Section 2.4 for additional information about the project). Therefore, it required an initial task of familiarization with the algorithms and processes developed in that project. An initial estimation of the tasks needed was performed, and the project was divided into two different blocks.

The first block was oriented to achieve an algorithm able to extract the geometric models from the road scenes, whereas the second block was centred on a study case over these geometric models: detection of tailgating. Consequently, the design and development of the system was in turn organized into two different subsystems, each of them assigned to a different group of tasks in the project. We next summarize the list of task identified in the project:

A. Documentation reading in order to familiarize with the context of the project.

B. Design of technical solution for the first block of the project (geometric models).

C. Implementation of first block of the project.

D. Test the algorithms developed during the first block using the provided dataset.

E. Design of technical solution for the second block of the project (tailgating detection).

F. Implementation of second block of the project.

G. Test the algorithms developed during the second block using the provided dataset.

H. Perform necessary experiments to check performance of the system.

I. Generate project manuscript's first draft.

J. Evaluate results of the experiments.

K. Update and complete project manuscript.

The tasks on this list were later analysed to evaluate the expected time to develop each task, resulting in the Gantt chart displayed in Fig. 67. However, during the project there were some deviations over the expected durations in the tasks. To illustrate this delay, Fig. 67 shows a red-dotted path, while the original expected path is displayed in green on the background. Every number denotes a week of work: 3 hours/day for 5 days.

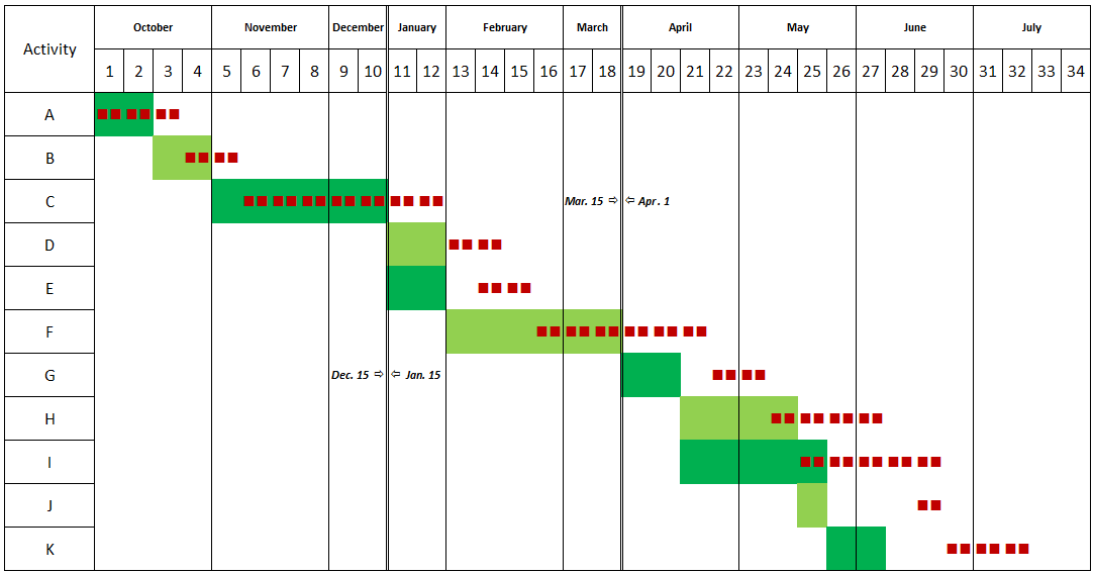


Figure 67. Gantt diagram with project development. Every square denotes a week. Expected time marked in green; Actual time marked in red dots.

From the diagram displayed in Fig. 67 yields that the total duration of the project was 32 weeks.

5.2 Development instruments

During the development of this project, several tools and resources were used. The main tool used during the development of the project was the software ***MATLAB***[4], with which the implementation of the system and the necessary experiments were made.

It should also be mentioned that, as this is a project included in a scholarship of the Signal Theory and Communications department at the Carlos III University, it was developed in its entirety in the facilities provided by the university. The office associated with the project is 4.2.A01. A computer located in this office, along with the computation servers of the department, were used for the development.

5.3 Budget information

This section presents the costs of the project. It includes the direct costs of the equipment used during the process, as well as the human resources cost associated to the participants in the project. Table 6 shows the human resources costs of the project.

Personel	Description	Cost /hour €	Hours*	Total cost €
Eduardo Pla Sacristán	Engineer	16.5	480	7920
Iván González Díaz	Engineer (Dr)	35	50	1750
Total cost				**9670**

Table 6. Human costs. Hours*: Associated to the project.

[4] MATLAB software: https://es.mathworks.com/

Table 7 shows the costs regarding the implementation of the project. It includes the costs of the equipment used in the process, taking into account the amortization factor for each piece of equipment. It was estimated that the amortization factor for the computer is 1 / 3, while for the monitors and other hardware external devices is 1 / 5, as the depreciation time is faster for computers.

Equipment	Cost (€)	Amortization factor	Total cost (€)
Department PC	650	1/3	216.67
Monitor Hyundai L72D	30	1/5	6.00
Monitor Hyundai L70S+	40	1/5	8.00
Keyboard Hewlett-Packard KB0316	20	1/5	4.00
Optical mouse Logitech MBT58	10	1/5	2.00
Office material	15	1	15.00
Total cost			**251.67**

Table 7. Equipment costs.

This yields the total cost for the project: ***nine thousand nine hundred and twenty-one euros and sixty-seven cents***. Table 8 shows the summary of human and implementation costs.

Item	Cost (€)
Human resources costs	9670
Equipment costs	251,67
Total cost	**9921,67**

Table 8. Total costs.

6 Conclusions and future work

This final chapter contains an analysis on the impact of the proposed system in its environment, as well as any further conclusions obtained from the experiments conducted. Moreover, a section devoted to future lines of work is included at the end of this chapter.

6.1 Conclusions

The first inference drawn from the termination of this project is that the current regulations for traffic safety need strong updates and improvements. There are plenty of factors that are not contemplated in the regulations of most countries due to a lack of methods to monitor and sanction it. However, this project has developed, with limited resources, a method to monitor one of these driving behaviours: tailgating.

This manuscript describes a system capable of determining general geometric models to transform traffic scenes to a top view where taking measures is feasible, regardless of the scene under analysis. This allows us to detect offensive driving behaviours that currently lack a proper preventive system.

As we have seen from the experiments performed, the deployment of the system is feasible in relatively general locations, as long as the camera perspective fulfils some simple requirements. However, as not all the views analysed during the experimental phase of this project fulfilled such requirements, they did not provide conclusive results. Still, 5 out of the 7 analysed locations provided a decent outcome in infraction detection, while only 2 of them were unusable. Some of the 5 valid camera locations suffered from intermittent problems, caused mainly by the influence of the illumination and the weather conditions in the process. Of course, these are all aspects that will be approached in future implementations of the system.

The data obtained from the tailgating detection system is very promising, as this driving behaviour clearly presents a threat to road safety, and yet it has not been regulated. This is partly due to the impossibility to monitor it and sanction it. With this automatic system, the possibility of accurate and affordable monitoring of tailgating behaviour is now closer. In this paper, a study case for tailgating detection has been proposed, but future lines of investigation could open a wide variety of possibilities. The system could be oriented to monitor many driving behaviours, such as reckless driving, changing two lanes at once, overtaking vehicles on the right, etc. This is possible thanks to the generality of the geometric models calculated.

However, this system is still incomplete concerning a real deployment as, for instance, the error rates are too high to be able to sanction drivers that show tailgating behaviour. It could be used, however, as a preventive tool, informing the drivers of their recklessness in those cases where tailgating is significantly clear. In the next section, other possible future real implementations will be discussed.

Therefore, we can conclude that, as a computer-vision-based system of detection and measurement, this system is, potentially, a strong alternative to current technologies, such as laser or radar based systems. This is because these systems usually cause great costs in infrastructure, and computer vision provides a less costly method of infraction detection and traffic monitoring. Additionally, as computer vision systems are usually formed by just digital camera and computers, they can be considered as passive technology and thus not permitting

detection by in-car devices. This will prevent drivers from avoiding infractions within monitored areas while still violating regulations outside them.

The main contributions of this project have been published at the Proceedings of the *Advance and Applications of Data Science and Engineer Workshop*, an international workshop organized by *Real Academia de Ingeniería* in Madrid (June 2016) [1].

6.2 Future work

As we have mentioned several times within this manuscript, this project is a pioneer investigation regarding whether it is feasible to take accurate measurements in traffic scenes using computer vision, in order to monitor driving behaviours that currently lack proper measurement and detection methods that would allow further regulation. Even though the results of this project are promising, the system here proposed is far from being a realistic enough method to implement real-time monitoring of driving behaviours such as tailgating. In this section, we propose some ideas to improve future results of this research or any related investigation.

6.2.1 Dataset improvement

Some of the requirements and restrictions specified in Section 3.1 are taken into consideration because the quality of the dataset is not optimal for this kind of processing. The dataset was intended for human monitoring, and it is therefore size-optimized, so the resolution of the stored images is rather poor and the videos are not oversampled. Improving the dataset would be the first step to obtain better results out of the system. Additionally, better quality images would require less adjusting processes such as morphological processing operations.

Therefore, improving the dataset will result not only in better results, but also in a reduction of the restrictions and requirements of the project.

6.2.2 Gathering of Ground Truth information

Future implementations of this system are expected to include the use of alternative technologies to obtain ground truths, such as radar, to complement the videos recorded. Radar technologies have shown great performance to measure distances and velocities, so the measures taken would definitely be more reliable and may help in the generation of an annotated dataset.

6.2.3 Automation of processes

The presented project currently relies partly on human interaction, as there is a module in the first sub-system, the *road lines classification module*, which allows us to choose a semi-automatic method that requires input from the user. There is also an automatic version for this method, but it has been proved that the performance of the semi-automatic method is better than that of the automatic method (see Section 4.2). An example of the limitations that the automatic method presents is the incorrect detection of certain DLs.

One of the objectives for the future is to achieve total automation of processes. The lesser the system has to rely on human interaction, the faster the implementation will be. Additionally, it would avoid the possibility of having human errors.

6.2.4 Homography generalization

There are some cases in which we have a reliable set of homography matrices for a scene at a certain one-hour video. However, the lighting in scenes that are very similar to this but at different hours does not allow us to obtain proper geometric models for them.

In the future, we are planning to use homography techniques to generalize the geometric models, so that the same geometric models can be used for different scenes if they are similar, even though one of them might not have produced practical models. This would consist in an additional transformation applied to the model-less scene so that the geometric models extracted from another scene could be applied to it.

6.2.5 Development of new applications over the geometric models

The geometric models developed during this project have been designed to be generic and usable in several situations.

For instance, the normalised top-view environment that is transformed around every Detection Area could identify a vehicle advancing on the right lane at a higher speed than a vehicle that is present on the left lane. From this, we could detect an irresponsible driving behaviour such as overtaking vehicles on the right. Another example could be measuring rapid changes in speed and location during a short period of time, in order to detect reckless driving.

Besides the geometric models, there are other contributions to this project that can be reused in future implementations. One example is the event specification module, as it was mentioned in Section 1.6.

References and bibliography

References

[1] PLA-SACRISTAN, Eduardo; GONZALEZ-DÍAZ, Iván; DÍAZ-DE-MARÍA, Fernando. Geometric models for video surveillance in road environments: tailgating detection. RAIng Proceedings of the advance and applications of data science and engineering, 2016, p. 43-48.

[2] AIRPLANES, Commercial. Statistical Summary of Commercial Jet Airplane Accidents. Worldwide Operations, 1959, vol. 2008.

[3] National Highway Traffic Safety Administration. [online]. Available: http://www-fars.nhtsa.dot.gov/Main/index.aspx. [Accessed 29-06-2016].

[4] UNIVERSIDAD CARLOS III DE MADRID. Técnicas robustas de visión artificial y su aplicación a los sistemas inteligentes de transporte para la mejora de la seguridad vial, la movilidad y la gestión del tráfico, Ref.: SPIP20141507, Entidad financiadora: Ministerio del Interior. Dirección General de Tráfico, 2015.

[5] DIRECCIÓN GENERAL DE TRÁFICO. Campañas. [online]. Available: http://www.dgt.es/es/la-dgt/campanas/. [Accessed 20-08-2016].

[6] LIVINGSTONE, Margaret; HUBEL, David H. Vision and art: The biology of seeing. New York: Harry N. Abrams, 2002.

[7] SZELISKI, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[8] HALL, Ernest L.; HWANG, J. J.; SADJADI, F. A. Computer image processing and recognition. En 1980 Los Angeles Technical Symposium. International Society for Optics and Photonics, 1980. p. 2-10.

[9] CASELLES, Vicent, et al. A geometric model for active contours in image processing. Numerische mathematik, 1993, vol. 66, no 1, p. 1-31.

[10] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. Nature, 2015, vol. 521, no 7553, p. 436-444.

[11] BODEN, Margaret Ann. Mind as machine: A history of cognitive science. Clarendon Press, 2006.

[12] EIKVIL, Line. Optical character recognition. citeseer. ist. psu. edu/142042. html, 1993. [online]. Available: http://s3.amazonaws.com/academia.edu.documents/33085443/OCR.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1474631164&Signature=cysOx0c0H5LpTVHGeOyLOao0mP8%3D&response-content-disposition=inline%3B%20filename%3DOCR_Optical_Character_Recognition_OCR_-O.pdf. [Accessed 23-08-2016].

[13] MOESLUND, Thomas B.; GRANUM, Erik. A survey of computer vision-based human motion capture. Computer vision and image understanding, 2001, vol. 81, no 3, p. 231-268.

[14] SILVEIRA, Margarida, et al. Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. *IEEE Journal of Selected Topics in Signal Processing*, 2009, vol. 3, no 1, p. 35-45.

[15] FOUNDATIONS OF DIGITAL IMAGE, Optical Character Recognition and Image Recognition. [online]. Available: http://teaching.paganstudio.com/digitalfoundations/?cat=7. [Accessed 07-07-2016].

[16] SMAOUI, Nadia; BESSASSI, Souhir. A developed system for melanoma diagnosis. International Journal of Computer Vision and Signal Processing, 2013, vol. 3, no 1, p. 10-17.

[17] READING UNIVERSITY, People Tracking for Visual Surveillance. [online]. Available: http://www.cvg.reading.ac.uk/projects/advisor/index.html. [Accessed 07-07-2016].

[18] SYSTEMATICS, Cambridge. Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. Final Report, Texas Transportation Institute. http://ops. fhwa. dot. gov/congestion_report_04/index. htm, 2005.
[19] KHALID, Shehzad; NAFTEL, Andrew. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. En Proceedings of the third ACM international workshop on Video surveillance & sensor networks. ACM, 2005. p. 45-52.
[20] AGHBARI, Zaher; KANEKO, Kunihiko; MAKINOUCHI, Akifumi. Content-trajectory approach for searching video databases. IEEE Transactions on Multimedia, 2003, vol. 5, no 4, p. 516-531.
[21] ALON, Jonathan, et al. Discovering clusters in motion time-series data. En Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. IEEE, 2003. p. I-375-I-381 vol. 1.
[22] AKÖOZ, Ö.; KARSLIGIL, M. E. Severity detection of traffic accidents at intersections based on vehicle motion analysis and multiphase linear regression. En Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 2010. p. 474-479.
[23] BARRIA, Javier A.; THAJCHAYAPONG, Suttipong. Detection and classification of traffic anomalies using microscopic traffic variables. IEEE Transactions on Intelligent Transportation Systems, 2011, vol. 12, no 3, p. 695-704.
[24] BALKE, Kevin, et al. Dynamic Traffic Flow Modeling for Incident Detection and Short-Term Congestion Prediction: Year 1 Progress Report. 2005.
[25] MITROPOULOS, George K., et al. Wireless local danger warning: Cooperative foresighted driving using intervehicle communication. IEEE Transactions on Intelligent Transportation Systems, 2010, vol. 11, no 3, p. 539-553.
[26] AHMED, Tarem; ORESHKIN, Boris; COATES, Mark. Machine learning approaches to network anomaly detection. En Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques. USENIX Association, 2007. p. 1-6.
[27] KOHAVI, Ron; PROVOST, Foster. Glossary of terms. Machine Learning, 1998, vol. 30, no 2-3, p. 271-274.
[28] CIPOLLA, Roberto, et al. Machine Learning for Computer Vision. Springer, 2013.
[29] MUNOZ, Alberto; MOGUERZA, Javier M. Estimation of high-density regions using one-class neighbor machines. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, vol. 28, no 3, p. 476-480.
[30] KÜHNL, Tobias; KUMMERT, Franz; FRITSCH, Jannik. Monocular road segmentation using slow feature analysis. En Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011. p. 800-806.
[31] SANTOS, Marcelo, et al. Learning to segment roads for traffic analysis in urban images. En Intelligent Vehicles Symposium (IV), 2013 IEEE. IEEE, 2013. p. 527-532.
[32] MELO, José, et al. Detection and classification of highway lanes using vehicle motion trajectories. IEEE Transactions on intelligent transportation systems, 2006, vol. 7, no 2, p. 188-200.
[33] CZAJEWSKI, Witold; IWANOWSKI, Marcin. Vision-based vehicle speed measurement method. En International Conference on Computer Vision and Graphics. Springer Berlin Heidelberg, 2010. p. 308-315.
[34] SHAH, Mubarak Ali; LIYANAGE, Janaka Pradeep. Homography-based passive vehicle speed measuring. U.S. Patent No 8,238,610, 7 Ago. 2012.
[35] KHAN, Saad M.; SHAH, Mubarak. A multiview approach to tracking people in crowded scenes using a planar homography constraint. En European Conference on Computer Vision. Springer Berlin Heidelberg, 2006. p. 133-146.
[36] JEPSON, Allan. Planar Homographies. [online]. Available: http://www.cs.toronto.edu/~jepson/csc2503/tutorials/homography.pdf. [Accessed 19-07-2016].

[37] HARTLEY, Richard; ZISSERMAN, Andrew. Multiple view geometry in computer vision. Cambridge university press, 2003.
[38] FISCHLER, Martin A.; BOLLES, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 1981, vol. 24, no 6, p. 381-395.
[39] HAYS, James Local Feature Matching. [online]. Available: http://www.cc.gatech.edu/~hays/compvision/results/proj2/tdao30/index.html. [Accessed 20-07-2016].
[40] GOYAL, Megha. Morphological image processing. IJCST, 2011, vol. 2, no 4.
[41] ELVIK, Rune. Why some road safety problems are more difficult to solve than others. Accident Analysis & Prevention, 2010, vol. 42, no 4, p. 1089-1096.
[42] Cambridge Dictionary of English. [online]. Available: http://dictionary.cambridge.org/es/diccionario/ingles/tailgate?q=tailgating. [Accessed 23-06-2016].
[43] Comisariado Europeo del Automóvil, Distancia de seguridad entre vehículos. [online]. Available: http://www.seguridad-vial.net/conduccion/reglas-circulacion/66-distancia-de-seguridad. [Accessed 28-05-2016].
[44] Ministerio de Fomento de España, Orden Ministerial sobre marcas viales. [online]. Available: http://www.fomento.es/NR/rdonlyres/56B5B61F-EEFA-4CB9-B050-CAA5E2172FDD/55741/1120100.pdf. [Accessed 21/07/2016].
[45] Ministerio de Fomento de España, Manual de criterios. [online]. Available: http://www.fomento.gob.es/NR/rdonlyres/8B19774A-F4B0-4553-B9D2-A7F69D11D729/103148/2010100.pdf. [Accessed 21/07/2016].
[46] BAÑON BLÁZQUEZ, Luis; BEVIÁ GARCÍA, José Francisco. Manual de carreteras. Volumen I: Elementos y proyecto. Alicante: Ortiz e Hijos, Contratista de Obras, SA, 2000.
[47] BENER, Abdulbari. The neglected epidemic: road traffic accidents in a developing country, State of Qatar. International journal of injury control and safety promotion, 2005, vol. 12, no 1, p. 45-47.

Additional bibliography

Besides the references stated before, other bibliography was consulted during the development of the project. Such bibliography includes:

- JIANG, Fan, et al. Detecting anomalous trajectories from highway traffic data.
- YU, Shih-Hao, et al. An automatic traffic surveillance system for vehicle tracking and classification. En *Scandinavian Conference on Image Analysis*. Springer Berlin Heidelberg, 2003. p. 379-386.
- HELALA, Mohamed A.; PU, Ken Q.; QURESHI, Faisal Z. Road boundary detection in challenging scenarios. En *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE, 2012. p. 428-433.

Appendix A. Project summary

A1. Introduction

Since the widespread growth of automobile use worldwide during the XX century, road safety has become a key issue for modern societies. Many strategies have been proposed to address this problem, such us regulated driving licenses, restrictive signaling and, more recently, speed monitoring through radar systems. The advances of technology have allowed traffic regulation entities to incorporate new preventive systems to real scenarios.

Among the broad family of technologies applied to road environments, computer vision systems have emerged as interesting solutions for certain problems due to their reduced cost and easy deployment. Video analytics and anomaly detection systems have been successfully implemented to properly detect some events, like traffic congestion, accidents or other anomalies that may occur in road scenarios. Detection is usually achieved by clustering or classification of object trajectories in videos. However, other lines of research also play a relevant role in anomaly detection, such as measuring microscopic traffic variables in real-time. Anomaly detection systems have also been improved by machine learning techniques. Computer vision systems are used, as well, for road segmentation purposes, with the intention of saving computational costs in subsequent video analysis. However, the use of computer vision is still limited when accurate distance or speed measures need to be computed, mainly due to certain limitations caused by lack of camera calibration, the camera viewpoint, occlusions or visual artifacts caused by variable illumination. Hence, radar techniques are still predominant to compute quantitative measures in road environments, as they provide reliable data in terms of speed or spatial location, often at the expense of a notably higher price. Even more, other driving behaviors (reckless driving, running stops/red lights, etc.) that clearly affect road safety cannot be supervised with radar systems and, as a result, they are neither regulated nor monitored.

This paper proposes a methodology to apply computer vision techniques to those scenarios that require computing quantitative measures related to general traffic or specific vehicles. For the sake of usefulness, the developed system will adapt to any road environment with little human interaction, thus being robust under many realistic circumstances: CCTV operators controlling traffic cameras may often change their viewpoint; external factors, such as wind or heavy rain, may also affect the position of the camera, etc. Our proposal would adapt to this unpredictable changes minimizing user's interaction.

In this paper, we have focused on one particular case of interest: vehicle tailgating. Although it represents one of the more frequent causes of traffic accidents, most countries do not present a clear regulation on this issue. Institutions in Spain, for instance, present recommendations on the distance considered to be safe in different situations, but no official regulations exist. However, much of the proposed processing pipeline could be easily applied to detect other offending driving behaviors without too much effort.

A2. Objectives

As it was stated in the previous section, computer vision has been traditionally limited regarding accurate measurements, as other technologies are more trustworthy in this area.

However, the improvement of the state-of-the-art and the enhancement of computer vision tasks via deep machine learning have allowed some alternatives to emerge from this area. The main objective, thus, of this project is to provide an alternative to regulate, monitor and even sanction traffic related incidents that were previously in hands of much more expensive technologies; and even aspire to accurately monitor certain behaviours that currently lack proper regulation.

In order to achieve this goal, a list of objectives concerning the project itself was drawn. It can be said that the main goal is to efficiently develop a system that is able to properly detect the desired driving behaviour, in order to prevent potential accidents in the future. With this in mind, we subsequently present three primary objectives:

- The system is able to automatically generate, when the necessary conditions apply, the geometric models used to effectively define the road.
- Using these models, the system is able to perform geometric transformations that allow us to take quantitative measures on a normalised (top-view) environment.
- From these measurements, the system is able to detect when tailgating behaviours are taking place on the scene.

These objectives are crucial, as the main historical problem with computer vision techniques has been the impossibility to take quantitative measures due to limitations on perspective, lighting or other unexpected artifacts. Some secondary goals are also established, like being able to produce a generic implementation that is adaptable for future lines of work; providing a robust method for homography estimation; or having a visualization interface of the results that is easy to interpret.

A3. Technical solution

It is worth noting that the current system for geometric modeling and tailgating detection builds over a previous system for video analytics in road environments previously developed by the Multimedia Processing Group of the Signal Theory and Communications department at the Carlos III University. Hence, some blocks are also available providing useful information such as road segmentation masks, background models for the scene, foreground masks indicating vehicles or even tracks associated with moving vehicles. In the remainder of this appendix, we will refer data or information coming from this system as *external*.

In order to create the design of the solution, the input and output of the system need to be determined. After a thorough analysis of the problem, we established a basic structure that can be summarized as follows:

- **Subsystem 1:** Geometric Models for Road Environments subsystem (GMRE). Devoted to define the road and its elements.
    - ***Input:*** Output from *external* system (background image defining the road scene).
    - ***Output:*** Geometric models of the road.
- **Subsystem 2:** Driving Violation Infraction Detection subsystem (DVID). Destined to detect tailgating behaviour.
    - ***Input:*** Geometric models from subsystem 1; output from *external* system (corrected video flow, background video flow, vehicle trajectory trackers).
    - ***Output:*** Infraction detection (tailgating).

The proposed system is based on the analysis of a video flow. The scene is analyzed frame by frame and a convenient set of features are extracted from it. The main goal is to avoid the need of human interaction with the system, by means of an automatic adaptation to new scenarios and road environments. As shown in Fig. A.1, the processing pipeline is composed of two main blocks, corresponding to the stated subsystems. The first block is in charge of obtaining the geometric models associated with the road environment. For that end, elements of interest in the road are detected and classified to support the subsequent determination of a geometric model. The computed geometric models will later allow the second block to obtain and interpret data from the scene in order to calculate accurate measures. In the case study of this paper, values for an estimation of distance between vehicles and their speeds are obtained and processed to detect tailgating events.

The following sections contain a summary of the functionality for each module displayed in Fig. A.1.

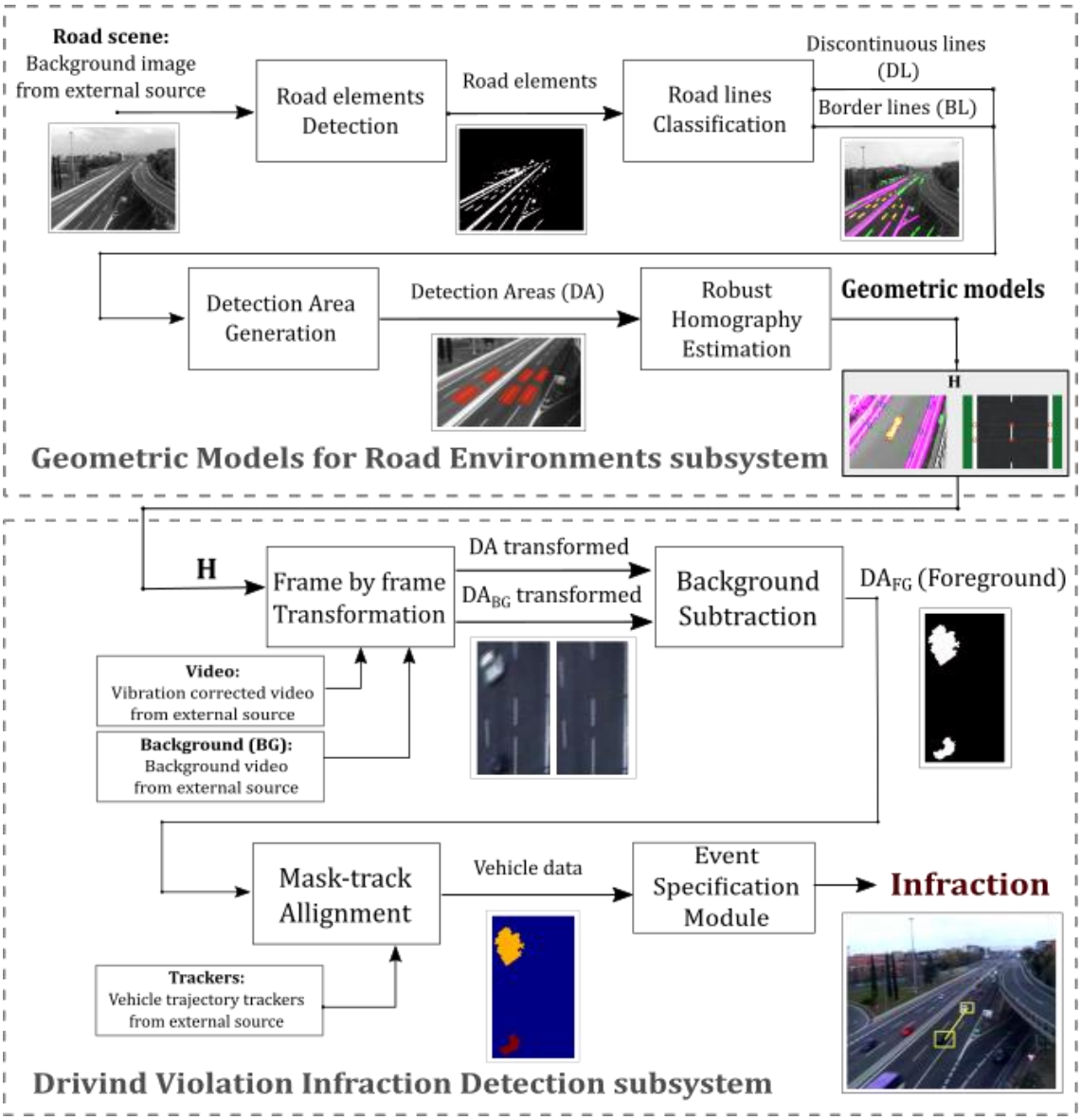


Figure A. 1. Processing pipeline of the whole system involving the GMRE subsystem (described in Section A3.1) and the Infraction Detection Subsystem (described in Section A3.2).

A3.1. Geometric Models for Road Environments subsystem (GMRE)

The first subsystem is destined to define the scene. In order to do so, we attempt to retrieve a transformation that models the original perspective of the image into a view in which we can identify events and take measures. For this, we need to have accurate knowledge about at least some of the elements of the road. Hence, this is precisely the starting line of the project.

A3.1.1. Road elements detection

The first step of the process is to detect the structuring elements of the road, namely, the set of white lines that define the limits of the road. For that end, an *external* input is used that consists of a background image of the analysed scene (Fig. A.2.a). Given the background image, a gradient-based analysis is carried out to detect the road lines that gives more importance to the gradient orientation in those parts of the image for which the gradient magnitude is more significant. A weighted histogram with the orientations is obtained and those objects that do not follow the main orientation of the image are discarded. Hence, this module is more effective detecting those edges corresponding to road lines.

A3.1.2. Road lines classification

Once a binary image identifying the objects of the image has been obtained (Fig. A.2.b), the next step is to determine which of these objects are discontinuous lines (DL), border lines (BL) or none of them. For this end, we have developed two alternative approaches:

- **Automatic method:** this system automatically decides which objects are DLs, which are BLs, and which are discarded from further processing, thus avoiding any human interaction. The process is based on restrictions on the detected candidates. In particular, constraints are applied to characteristics such as area, orientation, eccentricity and location, so those objects matching the expected criteria are marked as DL or BL, respectively.

- **Semi-automatic method:** This method requires some degree of human interaction. The automatic system will first identify candidate elements and, then, the operator is in charge of labelling them as DL, BL or others. Note that the user will only have to choose from an already processed selection of objects so that no drawing or precise definition of areas is required from the user.

As a result of any of these methods, an annotated image is obtained where DLs and BLs are identified. Fig. A.2.c shows a visual example of the output for this module, where different labels are determined over the original image (DLs: yellow, BLs: purple; other: green).

A3.1.3. Detection Area generation

Each one of the detected DLs will be then analysed separately, defining one Detection Area (henceforth referred to as ***DA*** in this appendix) per discontinuous line. The process in this module is very simple: it identifies the different DLs present on the image and it separates them into different DAs. These DAs may be later discarded if a homography matrix is not successfully extracted from it.

A3.1.4. Robust homography estimation

This module is the last of the first subsystem, and it will provide the final output of it, namely, a set of homography matrices (one per every non-discarded DA) that will transform each DA from its original camera viewpoint into a normalised top-view where quantitative measures can

be taken (Fig. 2.A.e). In order to estimate the geometric transform (homography matrix) relating both views, we need to identify at least four points in the original view whose coordinates are known in the top view. In our case, and as it is shown in Fig. 2.A.e, the proposed system uses a six-candidate-point model in an attempt to make the transformation more precise. However, it is noteworthy that not all of them have to be available to compute the transformation, thus providing a robust solution against visual artifacts.

Since the DL becomes the central element of our algorithm, the two points in its extremes should be always available. Starting from these points, the other four can be obtained, if possible, from the nearby objects of the scene (BLs, or other DLs in case of multi-lane motorways). For that end, two orthogonal lines are drawn from the first two points to connect the analysed DL with nearby BLs or DLs extensions. Fig. 2.A.d shows an example in which the six points are found in a DL and two BLs.

To estimate the homography, the Direct Linear Transformation (DLT) is used. The desired H is a 3x3 Homography matrix such that:

$$x_i' \times Hx_i = 0 \tag{1}$$

where $x_i = (x_i, y_i, 1)^T$ and $x_i' = (x_i', y_i', w_i')^T$ are the homogeneous coordinates of the key points in the original view and the normalized top view, respectively. From the eq. (1) it is easy to notice that vectors x_i' and Hx_i, which correspond to a matched pair of key points (original view and top view), must be parallel, i.e. they must only differ in magnitude by a factor w_i', no matter its value. Given the set of points (X, X'), the components of the matrix H can be obtained by transforming the eq. (1) into an homogeneous linear system represented as a matrix matrix-vector product, and applying Singular Value Decomposition (SVD) to it.

We would like to note that our approach is just an approximation and would be perfect only if the central DL and the nearby lines were parallel which, in general, is not completely true due to the projective view. However, it is well known that for planar surfaces (as the road) of small area with respect to their distance with the camera centre, projective transformations can be successfully approximated by affine transformations which do not break the parallelism between lines. Under this hypothesis, the use of orthogonal lines from the DL to the nearby lines (DLs or BLs) will be precise enough.

It should also be mentioned that we have developed a method to add robustness to the point detection. The system is based on the RANSAC (RANdom SAmple Consensus) method. The method here developed is not random, but it chooses the optimal combination of detected points in order to obtain the highest possible quality for the homography. Thanks to this method, we are able to discard possible outliers that may affect negatively to the resulting transform.

Figure A. 2. Visualization of different stages of the GMRE subsystem.

A3.2. Driving Violation Infraction Detection subsystem (DVID)

The second subsystem has the main task of obtaining the final output of the system: tailgating detection. For this, it makes use of another four modules, which are briefly explained below.

A3.2.1. Frame by frame transformation

As opposed to the GMRE subsystem, this second subsystem analyses the scene through the video, this is, frame by frame. Hence, it is important to highlight that the process described in this section is repeated for each frame of the video. The input video was previously stabilized by the *external* system to compensate for vibrations caused mainly by the wind. Our system also uses a second input coming from the *external* system, which corresponds with a background model (road image without vehicles and other moving elements, as shown in Fig. A.2.a) that is updated at each frame.

This first module transforms the DAs using the corresponding homography matrices obtained in the first block. The transformation is performed in both the original video and the background video, yielding the DAs with and without vehicles. This is illustrated in Fig. A.3.b.

A3.2.2. Background subtraction

This module performs a simple image subtraction between the current top view and its corresponding background. This difference is then thresholded to obtain a binary mask representing the foreground (FG) of the current DA (Fig A.3.c Left). This foreground DA gives us an idea of the vehicles present in the DA at the time corresponding with the analysed frame. However, the system does not have a notion of each vehicle as an independent object yet. Instead, it considers just two types of regions (vehicle area and no-vehicle area).

A3.2.3. Mask-track alignment

The next step aims to identify individual vehicles in the scene. To achieve this, another input is gathered from the *external* system: the tracks of the vehicles in the road. Each track is a sequence of spatial locations and shapes (encoded as bounding boxes) that define the trajectory of a vehicle in the scene. The tracks are obtained online, so that at each frame we can get the bounding boxes associated with every vehicle in the frame. However, let us note that the goal of the tracking subsystem is to obtain an approximate location of a vehicle, lacking the precision needed to measure speeds or distances between vehicles.

This information about trackers is then paired with the foreground masks in the current DA so that each connected component is aligned with one vehicle in the tracker database. The output of this module is an image in which every vehicle present on the region of interest is identified (Fig. A.3.c Right).

A3.2.4. Event specification

In the particular case of tailgating, for those pairs of vehicles moving in the same lane, the system pursues measuring the distance separating them as well as their speed. To this end, this module has been in turn decomposed into three basic sub-modules:

Event detection: it detects when two or more vehicles are present in the top view of the current DA and evaluates whether they are driving in the same lane or not. If the system detects at least two vehicles in the same lane within the area of the DA (Fig. A.3.c), an event is created.

Event measurement: once the event is detected, this sub-module measures the considered magnitudes. For the case of tailgating, these are the distance between both vehicles and the speed at which the rear vehicle is moving (it only takes into account the speed of the vehicle behind, as it is the one causing the violation).

In order to obtain such measurements, the system establishes key points that define each of them. In the case of the distance, the calculation is based on the two points belonging to adjacent vehicles that are closer to each other. This is: the top of the vehicle at the bottom of the image and the bottom of the vehicle at the top of the image. As the domain in which the event is detected is normalized, we can neglect the horizontal coordinate and simply perform a subtraction between the vertical coordinates of each point.

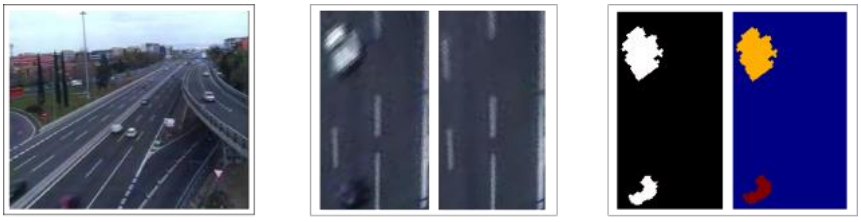
It is worth mentioning that the distance between vehicles at a given moment does not require information from previous frames, so it is only measured when an event is detected. In contrast, the speed of the vehicles is obtained for every vehicle passing through a DA, even if the vehicle does not need any further processing (there is no other vehicle in the same lane). This is done due to the need of at least two frames to compute the speed. Hence, the system requires measuring and updating the speed of every vehicle passing through a DA. For this, it takes note of the front point of each vehicle, i.e. the most forward point belonging to the analysed vehicle, and the frame in which is detected it. From this set of points, it later estimates a motion vector that will be used to calculate the speed.

Event classification: it takes as input the values that define the event (in this case, the speed and the distance), and it estimates the danger of the situation. Depending on this danger factor, the system decides if tailgating exists.

The specific factor used to decide on the danger is the time of impact t_i:

$$t_i = \frac{d_v}{v_r} \tag{2}$$

where the parameter v_r is defined as the speed of the rear vehicle and d_v is the distance between adjacent vehicles. The time of impact represents the time needed by the rear vehicle to impact the front one in case of a sudden stop. Assuming that both vehicles need the same time to stop once the brake pedal is pressed, then, time of impact is the time that the driver of the rear vehicle has to react to a sudden break of the previous vehicle.



(a) (b) (c)

Figure A. 3. Visualization of different stages of the DVID subsystem.

A3.3. System output

The final system output is a set of pairs of vehicles driving in the same lane with their corresponding times of impact t_i. By setting up an absolute threshold over t_i, one could determine whether or not tailgating is taking place. Unfortunately, due to the lack of official

regulation, this threshold is not specified and only recommendations can be found in the literature. Fig. A.4 shows an example for the visualization of the system output, in which an infraction is detected.

Figure A. 4. Infraction detection. Example of visualization.

A4. Experiments and results

This section presents the dataset used to develop the project and to perform the experiments, also briefly explained in this section. Additionally, we will comment on the results of these experiments.

A4.1. Dataset

We have assessed the performance of the proposed method with different videos recorded by the CCTV system of the Spanish Traffic management Administration (DGT). The videos show real scenarios with varying traffic situations. As it has been stated before, this work corresponds with an on-going research, and so the results are still preliminary.

The current dataset consists of seven locations. For each location, we have videos between 8 a.m. and 17 p.m. During this period of time, each camera changes its point of view at least once. As a result, we have obtained as much as 80 different background images in which the geometrical models can be calculated. Some examples are shown in Fig. A.5.a.

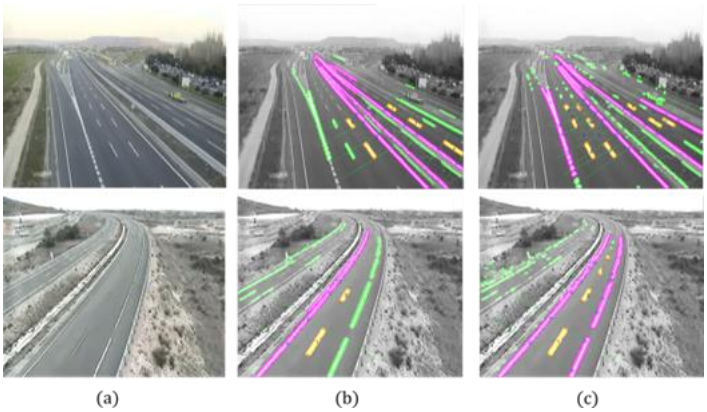


Figure A. 5. Example of background image (a) and performing results of automatic (b) and semi-automatic (c) methods for two camera locations.

A4.2. Experiments on road line classification

As it was explained in Section A3.1.2 of this summary, this system has two different methods that can be used to classify the elements of the road into the selected categories (DL and BL). In order to determine the performance of the considered methods, a comparative study was performed.

	Automatic	Semi-automatic
Total	79	142
Excellent	45	109
Acceptable	11	30
Defective	3	3
Incorrect	20	0

Table A. 1. Total number of homographies calculated. Human error absence is assumed for semi-automatic method.

Table A.1 shows a relation between the number of DAs found in the whole sample space for each method. From a strictly objective point of view, we can confirm at first glance that the performance of the semi-automatic method is better, as the total number of homographies found with this method is almost twice the number found with the other. A subjective study was also conducted to evaluate the quality of the obtained transformations, yielding positive results for both methods, but again better for the semi-automatic method. In this subjective study, four labels were created to define the quality of the homography performed: **excellent** (very good quality), **acceptable** (good quality), **defective** (unusable quality) and **incorrect** (not performed on a correct DL).

As we can see, the automatic method does not always detect correct homographies, as it does not distinguish actual DL from incomplete DL or BL. The semi-automatic method deflects this issue with user involvement in the process. The automation of this kind of discrimination is reserved for future lines of work.

A4.3. Theoretical study on the precision of the measurements

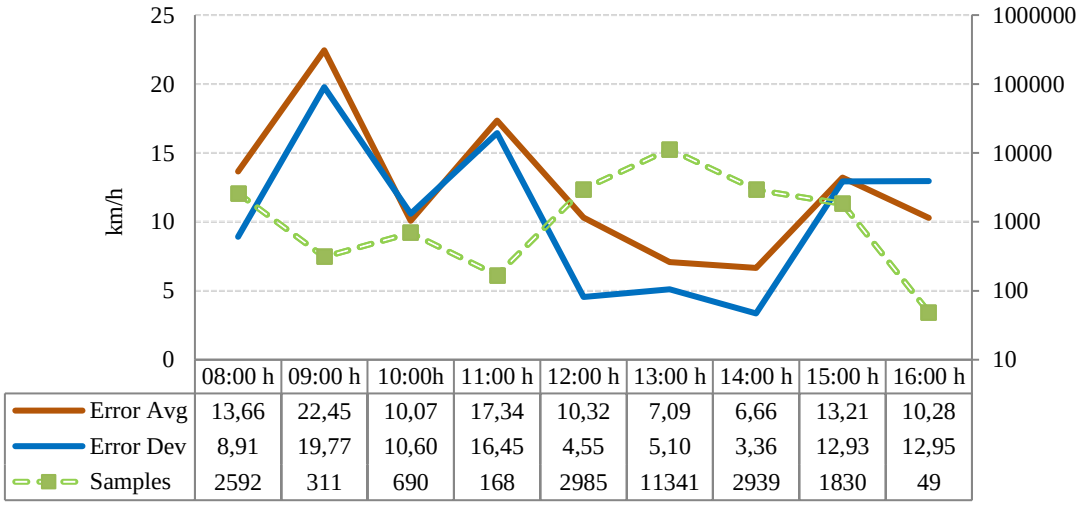
It is important to notice that the system has external limitations that cannot be avoided by any image processing techniques. The video resolution (spatial and temporal) affects greatly on the results obtained by the system. As any digital system, this one has errors due to quantization. In order to understand how much the system is affected by this matter, we need to study the uncertainty of the measures we are taking. For this, a theoretical study on this uncertainty was performed.

In this study, it was stated that the error varies significantly from one DL to another, as the resolution for DLs that are farther from the camera is worse. In a general scene, the error caused by this uncertainty can go from around 10 to 25 km/h. As we can see, the error is high even for the best cases. Moreover, this error will be increased by the uncertainty caused by the detection of points, transformations and further processing performed into the images. Therefore, this would be unacceptable for a real system, especially taking into account that this particular error could be greatly reduced easily. According to this study, in the same conditions but with a video resolution of 1080p and 60 fps, the error range would be drastically reduced, reaching errors even one magnitude lower (around 1.5 km/h).

A4.4. Statistical study on the quality of the measurements

Direct observation of the results over various traffic sequences has allowed us to check that, when the majority of the system requirements are satisfied, the system successfully detects those cases where tailgating is very clear. Nevertheless, this evaluation is rather subjective, and we still lack an objective method to evaluate the quality of our measurements. Therefore, a study about the measurements taken was performed, in order to obtain objective and interpretable results.

The results obtained depend strongly on the camera location under analysis, as some of the camera locations produce worse results or even no results at all. However, the most significant cause of the variation in the results (given a certain scenario) is the external conditions (lighting, time of day, weather). In Fig. A.6 we can observe that the ideal hour range for detection is from 12 h to15 h, whereas the rest of the day usually generates worse results.



	08:00 h	09:00 h	10:00h	11:00 h	12:00 h	13:00 h	14:00 h	15:00 h	16:00 h
Error Avg	13,66	22,45	10,07	17,34	10,32	7,09	6,66	13,21	10,28
Error Dev	8,91	19,77	10,60	16,45	4,55	5,10	3,36	12,93	12,95
Samples	2592	311	690	168	2985	11341	2939	1830	49

Figure A. 6. Error analysis by hours showing the average error (red), the error deviation (blue) and the number of samples (green) each hour. The right vertical axis marks the number of samples, while the left vertical axis marks the speed in km/h. Please note that the character ‘,’ is used to separate the integer from the decimal part.

A5. Conclusions and future work

We have developed a system capable of determining general geometric models to transform traffic scenes to a top view where taking measures is feasible, regardless of the scene under analysis. This allows us to detect offensive driving behaviours that currently lack a proper preventive system. Additionally, as this method only makes use of computer vision, it is passive technology, meaning that it is undetectable for the driver (unlike radar related technologies).

The data obtained from the tailgating detection system is very promising, as this driving behaviour clearly presents a threat to road safety, and yet it has not been regulated. This is partly due to the impossibility to monitor it and sanction it. With this automatic system, the possibility of accurate and affordable monitoring of tailgating behaviour is closer. Here, a study case for tailgating detection has been proposed, but future lines of work could open a wide variety of possibilities. The system could be oriented to monitor different driving behaviours, such as reckless driving, changing two lanes at once, overtaking vehicles on the right, etc.

The main contributions of this project have been published at the Proceedings of the *Advance and Applications of Data Science and Engineer Workshop*, an international Workshop organized by *Real Academia de Ingeniería* in Madrid (June 2016).