INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Automatic Vehicle Recognition System

An approach using car rear views and backlights shape

David João Adão dos Santos

Dissertation for obtaining the degree of Master in

Electrical and Computer Engineering

Jury

Supervisor: Prof. Paulo Correia

President: Prof. António Topa

Members: Prof. Fernando Pereira

September 2008

Acknowledgements

The author would like to thank everyone who contributed to the whole project's development with special regard to its supervisor, Prof. Paulo Correia, and the Image Group of Instituto de Telecomunicações for the hospitable treatment.

Thanks also go to Family and friends: Ana Margarida, João Santos, Maria João, Filipa Mateus, Ricardo Santos, Jorge Fazenda, Rodrigo Piedade, Patricia Belchior, Mariana Almeida, Luis Grancho and Gisela Roncon for all the support and patience.

.

Abstract

In modern society, due to the high crime rates and high number of traffic accidents the feeling of insecurity and threat is increasing. The need for the establishment of defence and prevention mechanisms has encouraged studies to develop an automatic recognition systems, which should for example, have the ability to recognize vehicles at a certain distance.

The work reported here focuses on the implementation of two different methods, both relying on the analysis of vehicle external features, to identify the vehicle's manufacturer and model. The first method evidences the external shape of the vehicle, especially its dimensions and edges, while the second considers features computed from the vehicle lights, notably the orientation, eccentricity, position, angle with vehicle license plate and its shape contour. Both methods are then combined in order to obtain the proposed automatic vehicle recognition system.

Experimental results indicate that vehicle external features can be exploited as a feasible technique for automatic vehicle recognition showing a correct recognition rate of 88% on a database acquired at the IST parking.

Keywords

Image Processing, Automatic Vehicle Recognition, Vehicle Features Extraction, Vehicle Shape, Color Extraction.

Resumo

Na sociedade actual, devido aos elevados índices de criminalidade e de acidentes rodoviários tem-se verificado um aumento significativo de insegurança das populações. A necessidade de criar mecanismos de defesa e prevenção tem incentivado cada vez mais o desenvolvimento de sistemas de reconhecimento automático, que deverão por exemplo, ter a capacidade de reconhecer veículos a uma determinada distância.

Este projecto consiste na implementação de dois métodos distintos de reconhecimento de veículos, para identificar a marca, modelo e cor do veículo através da análise das suas características exteriores. O primeiro baseia-se num estudo exaustivo da forma exterior de um veículo, especialmente as suas medidas e contornos. Por outro lado, o segundo método apoia-se na análise cuidada de um dos componentes integrantes de qualquer veículo, os seus faróis. Os faróis de cada veículo são então caraterizados em termos de orientação, excentricidade, posição no veículo, ângulo relativo à matricula e respectiva forma. Finalmente ambos os métodos são combinados de forma a obter o sistema de reconhecimento automático de veículos proposto.

Após um número elevado de testes provou-se que as características exteriores de qualquer veículo podem ser utilizadas como forma de os reconhecer, obtendo o sistema implementado uma taxa de reconhecimento correcto de 88%, usando uma base de dados criada no parque de estacionamento do IST.

Palavras-chave

Processamento de Imagem, Reconhecimento Automático de Veículos, Extracção de Características dos Veículos, Forma de Veículo, Extracção de Cor.

Table of Contents

List of Figures

List of Tables

List of Acronyms

BEMS	Binary Edge Map Similarity
CCD	Charge-Coupled Device
ES	Eccentricity Similarity
GUI	Graphical User Interface
HSV	Hue, Saturation and Value
HG	Hypothesis Generation methods
HV	Hypothesis Verification methods
IST	Instituto Superior Técnico
LBSS	Lights Binary Shape Similarity
LDSS	Light external shape's distance signal Similarity
LPPS	License Plate Position Similarity
LPS	Lights Position Similarity
OS	Orientation Similarity
PCA	Principal Component Analysis
RGB	Red, green and Blue
SIFT	Scale Invariant Feature Transform
TR	Threshold value
VBSS	¾ of Vehicle Binary Segmentation Similarity
VDSS	Vehicle external shape's Distance Signal Similarity
WHCS	Width/Height Coefficient Similarity

Chapter 1

Introduction

1.1 Overview

Automatic vehicle detection and recognition has become, in the last years, an important subject of study. Many related applications that have been developed, such as self-guided vehicles, driver assistance systems, intelligent parking systems, measurement of traffic parameters and probably the two most important, related with surveillance problems for fighting crime and preventing terrorism, and accident prevention, continue to suggest that a vehicle recognition system is a good manner to help some different areas such as safety, intelligent transport systems and traffic management.

One of most common approaches to vehicle detection is using vision-based techniques to identify vehicles from images or videos. However, due to the wide range of vehicles colors, sizes, orientations, shapes, and poses, developing a robust and effective vision-based vehicle detection system is very challenging.

On the other hand, due to the uncountable number of different vehicle manufacturers and models in our market, the difficulty of building an efficient and complete vehicle recognition system is understandable, and so the method here proposed is a "*work in progress*" approach.

1.2 Motivation and Objectives

This project aims to develop an automatic vehicle detection and recognition system based on vehicle's external features, which should be robust enough to detect color, vehicle manufacturer and model. The main advantage over other types of vehicle recognition systems comes from the fact that vehicle external features can't be easily changed, like for example its license plate, since they are components of the vehicle.

Such system could be used in many different situations, for example, in a supermarket parking, or something similar, in order to automatically track all the traffic information during a desired period of time or even be part of a system that helps each supermarket client to find his previously parked vehicle. It could be also a great update to systems already in use like radar controls and vehicle toll controls in motorways. Most of the vehicle recognition systems in use exploit vehicle license plate as a *keypoint* to recognition purposes, therefore to prevent crimes where vehicle license plate could be false, the proposed system is of added value, verifying whether a recognized license plate correctly matches with respective vehicle color, manufacturer and model.

The method being developed uses video sequences of vehicle rear views, captured with a static camera, and intends to detect and recognize vehicles. To improve the system surveillance capabilities all the required procedures should be done with the aim of a real-time application, or close to real-time. Even if that is not possible, it can be developed in order to work as, for example, a traffic control system to vehicle parks, motorways or likewise.

1.3 Contributions

Knowing that vehicle's external features are particular to each model, the automatic vehicle detection and recognition system here presented is based on two novel vehicle external feature extraction methods developed by the author: *Vehicle Shape Features* and *Vehicle Lights Features* that are then combined in order to obtain a robust system.

The first method evidences the external shape of the vehicle, especially its dimensions, edges, and shape contour, while the second considers the vehicle's back lights features by following several ways to classify them: their orientation, eccentricity, position, angle with vehicle license plate, and their shape contour.

The main contribution of this Thesis is the development of these two novel methods that combine, in an original way, a set of vehicles features usually referred to in articles related to vehicle recognition. The achieved results prove that these methods can be exploited as feasible techniques for automatic vehicle recognition, and demonstrate the important contribution of this Thesis, principally since the vehicle recognition issue is still at its infancy.

1.4 Structure of this Report

Along this report the vehicle recognition problem is addressed; its structure consists in 6 main chapters. Chapter 1 provides an introduction to the whole project. Chapter 2 reviews the state of the art, providing a short description of the main published algorithms for vehicle detection and recognition. Chapter 3 provides a description of the proposed detection and recognition strategies, focusing on the details of the algorithms' implementation. Chapter 4 discusses the software implementation, also providing instructions on how to use the developed graphical user interface. An extensive set of experiments were made to evaluate the method's performance. The analysis of the results, achieved under various test conditions, is presented in Chapter 5. Finally, Chapter 6 presents the conclusions drawn from the work developed and highlights possible directions for future work.

Chapter 2

State of the Art

Since the 1970s, the problem of vehicle detection/classification has been studied by several researchers and now, many studies show that it is possible to detect if a given object is a “vehicle” or a “non-vehicle”. On other hand, not many existing solutions to recognize vehicle manufacturers and models are available in the literature.

This project is divided into two main parts: “Vehicle detection” and “Vehicle recognition”. As such, this review of the state of the art follows a similar structure.

2.1 Vehicle Detection

In recent years, we have seen a number of systems being proposed for the detection, segmentation and tracking of vehicles [1, 7, 8, 22], where robust and reliable vehicle detection is the first step. (*Video tracking* is the process of locating a moving object in time using a camera).

The majority of methods reported in the literature follow two main steps [1], the Hypothesis Generation methods (HG), where the locations of possible vehicles in an image are hypothesized and Hypothesis Verification methods (HV), where tests are performed to verify the presence of vehicles in the location previously hypothesized, resulting in a “vehicle” or “non-vehicle” automatic classification system.

Classification of vehicles opens up a wide range of research, for example, vehicle or non-vehicle, shapes, sizes, categories, etc. It is even more difficult when vehicle classification includes real-world images, for example, traffic scenes, occlusions, shadows, camera noise, changes in the lighting and weather conditions.

2.1.1 Hypothesis Generation Methods

Various Hypothesis Generation (HG) methods have been proposed in the literature, which can be classified into one of the following three categories:

- Knowledge-based methods;
- Motion-based methods;
- Stereo-based methods.

Each of these methods is detailed below.

Knowledge-based methods

Knowledge-based methods employ a priori knowledge of certain features of vehicles to hypothesize their location in an image.

An inherent problem with feature extraction approaches is that they are sensitive to local or global image variations (pose changes, illumination changes, and partial occlusion).

A list of some of the most important vehicles' features often used is:

- ***Texture and Color***: Due to the similarities of all vehicles (like windshields, rear bumper and vehicle license plate) their presence in an image causes a local intensity change that follows a certain texture pattern different from other areas [2], such as tree shadows or likewise. The smudge on the road caused by a presence of a vehicle can be also used as a cue for vehicle detection [7]. Image texture characteristics are often described by estimating the roughness of the images. Entropy is another measure used for texture detection [1], as illustrated in Figure 2.1.
  Also color information, which is very helpful to follow lanes/roads [3], can be used to improve the segmentation of vehicles from background.

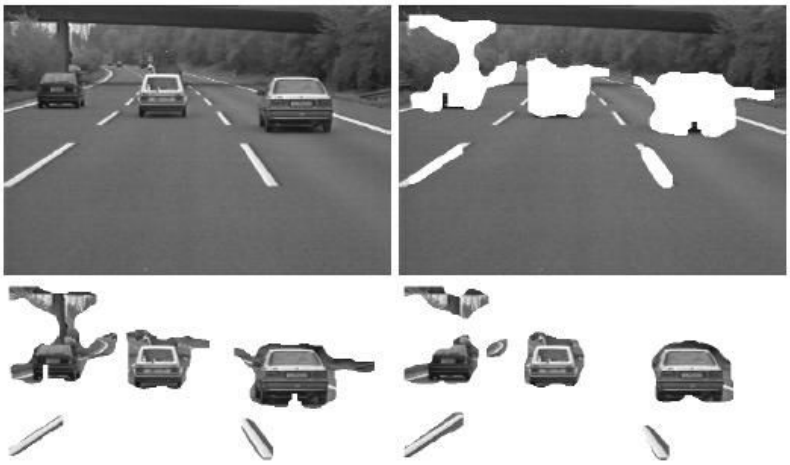


Figure 2.1 - Image segmentation based on local image entropy [2].

- ***Shadow***: By investigating image intensity, it was found that the area underneath a vehicle, its shadow, is distinctly darker than any other areas on an asphalt paved road [1], thus being a major aspect to exploit in vehicle detection. However, the intensity of the shadow depends on the illumination of the image, which in turn depends on weather conditions. Therefore the main constrain of using this feature is the estimation of a correct threshold to find what is the true shadow of the vehicle – see Figure 2.2. Many approaches reported in the literature use this feature [7], [8], [9], [10].

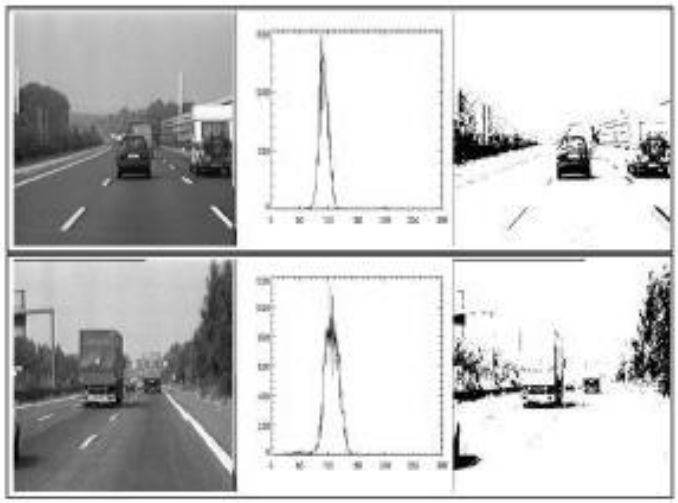


Figure 2.2 - Gray value's histogram and respective thresholded images [9].

- ***Geometrical Features*:** Depending on which vehicle view is selected different approaches can be considered, but in general, especially rear/frontal views, contain many horizontal and vertical structures, such as rear-widows, bumper, etc. Therefore, the detection of vertical and horizontal edges is a strong cue for hypothesizing vehicle presence, with which it is possible to localize a bounding box containing the vehicle [7], [8] – see example in Figure 2.3.
  Corners are another type of geometrical features that can be used for vehicle detection, as vehicles in general appear in images with a rectangular shape containing four well marked corners (upper-left, upper-right, lower-left and lower right) [21]. Other geometrical features used in some approaches are the vehicle's dimensions [10]. However, this has a great handicap due to the fact of depending on the distance of the vehicle to the camera, and the vehicle model itself, as dimensions can change introducing great uncertainty in the final result.

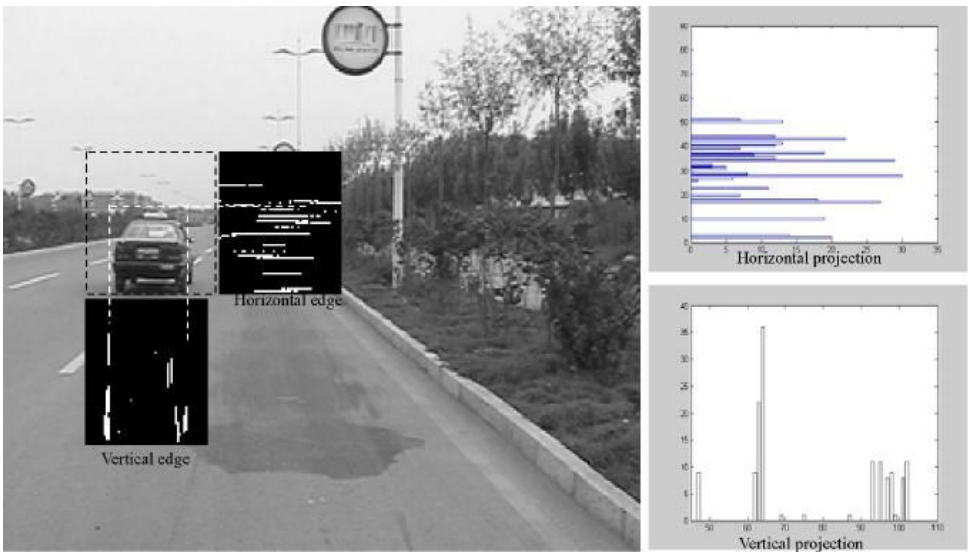


Figure 2.3 - Vehicle location based on edge information [7].

- ***Symmetry***: Although the modern architectural fashion tries to change it, one of the main characteristic of man-made objects is symmetry. The same happens with vehicles, symmetry is a typical vehicle's characteristic, which includes profile symmetry, binary value profile symmetry [11], grayscale symmetry [12], contour symmetry, horizontal symmetry, vertical symmetry, S Component Symmetry in HSV Space [12], etc. Some examples of symmetries are shown in Figure 2.4. This observation has been used as a cue for vehicle detection in several studies in computer vision [7], [13], [22]. This symmetry can be estimated for all previous features being the most commonly exploited geometrical feature. An important limitation of computing symmetrical characteristics in vehicle detection is the noise present in the images.

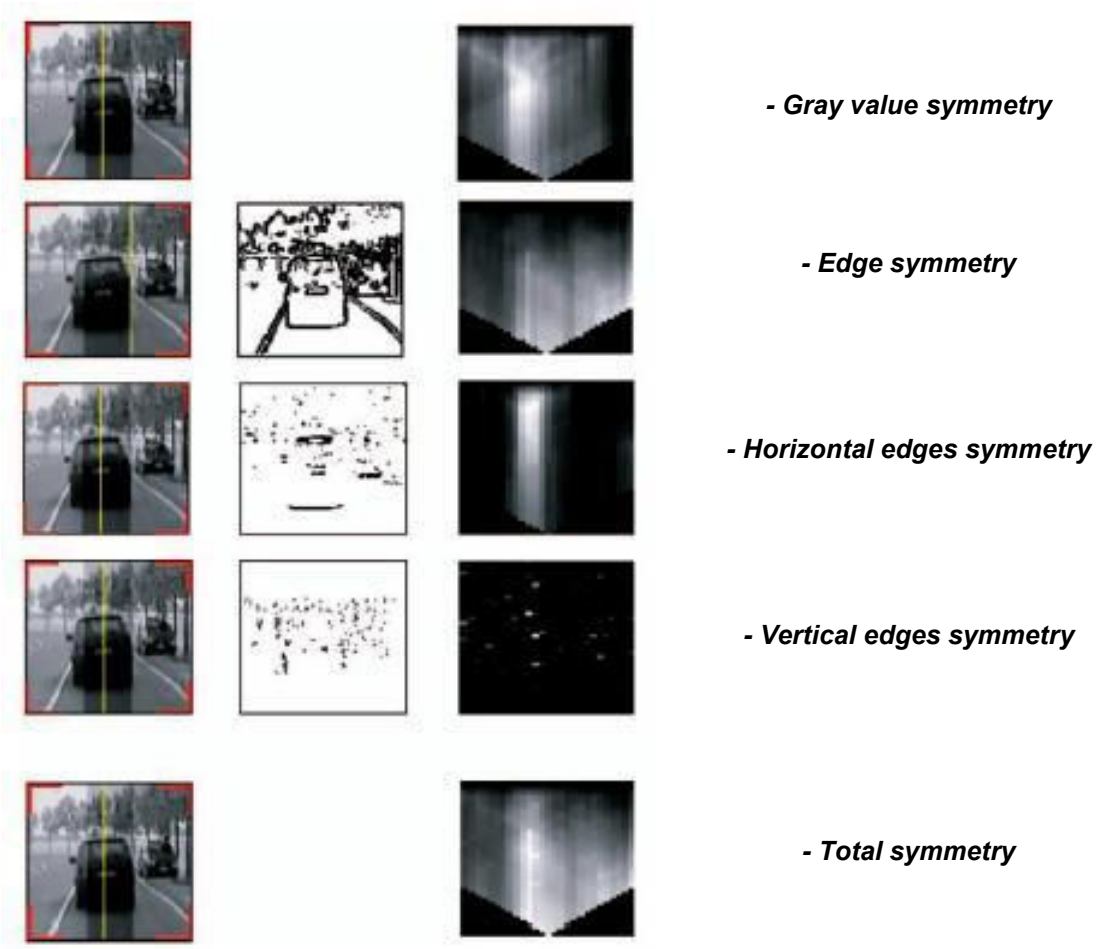


Figure 2.4 - Examples of computed symmetries [22].

- ***Vehicle Lights***: All vehicles, excluding some motorcycles, have a pair of back lights, most of the time with the same color, "red", and shape and size varying from model to model - see example in Figure 2.5.

Although this feature has a disadvantage comparing to the others, due to the fact that depending on the outside illumination, lights could be on or off, this fact has to be taken into consideration in any detection system that relies on the vehicle lights detection.

Figure 2.5 - Examples of vehicles pair of back lights.

- ***License Plate***: It's known that by law all vehicles need to circulate with a valid license plate that has the same specific characteristics (size, position, colors, shape, etc) - see Figure 2.6. Therefore, it's easy to understand that if the moving object has those similar characteristics it may be a license plate and consequently part of a moving vehicle.

Figure 2.6 - Portuguese license plate examples.

The above listed features may be used alone or in combination to aid in the automatic detection of vehicles in images and videos, when using knowledge-based methods.

Motion-based methods

When dealing with videos containing moving vehicles, motion-based detection methods can be employed. These are statistical methods that do not require sophisticated prior models. The detection of vehicles can be done by analyzing the statistics of the spatiotemporal patterns generated by the vehicle motion in the images [23].

All the knowledge-based methods described so far, use spatial features to distinguish between vehicles and background. Motion-based methods may use, as a cue, the relative motion obtained via calculation of the optical flow created by the motion of the vehicles [4], [5], as the pixels on the images appear to be moving due to the relative motion of the vehicles - see Figure 2.7. Another example of motion based methods, applicable when using static cameras, is the well-known "*Median method*", where the median value for each pixel is estimated for a time interval, allowing to distinguish moving from static objects in a sequence of images, and thus to easily segment them from the background.

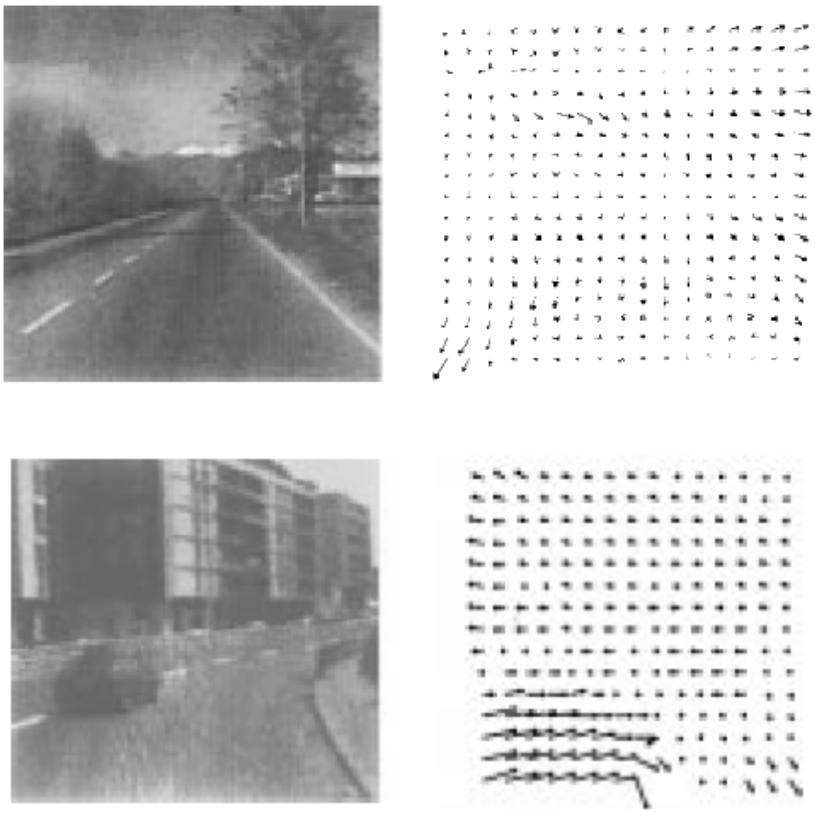


Figure 2.7 - Optical flow of road scene examples [4].

Stereo-based methods

Another type of Hypothesis Generation methods, are stereo-based methods, which make use of stereo information for vehicle detection.

Although not so commonly found in the literature, it is possible to highlight two different methods:

- ***Disparity Map*:** A disparity map is formed with the disparities of corresponding pixels of stereo images points. Once the disparity map is estimated, all the pixels with a disparity value higher than a defined threshold are accumulated in a disparity histogram. If an obstacle (vehicle or something moving) is present then a peak will occur at the corresponding histogram point. According to Franke et all [6], each pixel should be classified into a category (vertical edge pixels, horizontal edge pixels, corner edge pixels, etc.) based on the intensity differences between the pixel and its four direct neighbours.

- ***Inverse Perspective Mapping*:** Inverse Perspective Mapping is defined by the following procedure: If we consider a point *p* in the 3D space, perspective mapping implies a line passing through this point and the center of projection *N*. For a point p_I', (point in the image plane) the associated ray is traced through *N* towards the horizontal plane. The intersection of the ray with the horizontal plane is the result of the Inverse Perspective Mapping -see Figure 2.8. Now, assuming a flat road, it's easy to understand that, composing the Inverse Perspective Mapping, the horizontal plane is mapped onto itself, while elevated parts of the image appear distorted. Comparing both images of the stereo vision it's possible to find contours of objects above the ground plane.

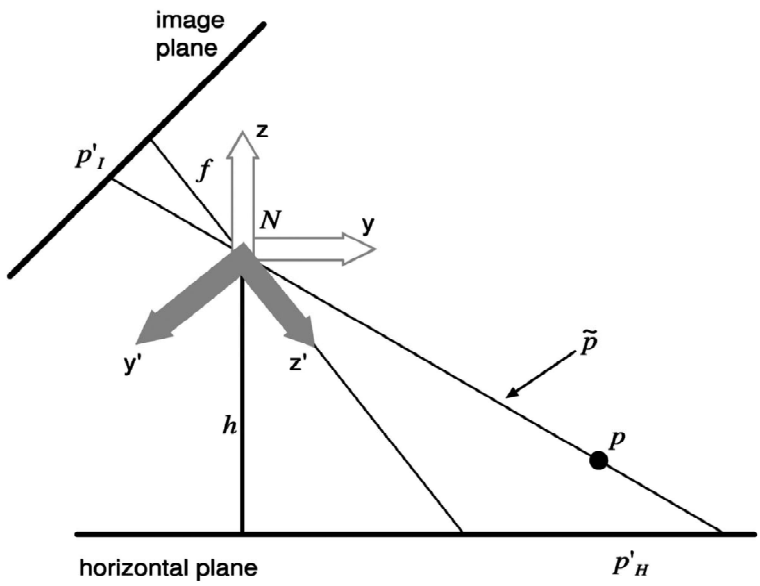


Figure 2.8 - Geometrical perspective mapping [1].

Most existing published vehicle detection methods are combinations of the techniques described above, combining knowledge and motion-based methods, and eventually also stereo information if available. In Section 2.1.3., a list and a brief description of some methods proposed in the literature is presented.

2.1.2 Hypothesis Verification Methods

After generating a set of vehicle location hypothesis, using HG methods, those hypotheses need to be verified, using Hypothesis Verification (HV) methods.

HV methods include tests to verify the presence of vehicles in the locations previously identified and can be divided into two main categories:

- ***Template-based:*** these methods use predefined patterns from vehicles and perform some correlation between the selected image locations and the predefined template(s). These templates are often defined by a mix of the features previously described in knowledge-based methods (HG methods);

- ***Appearance-based:*** these methods learn the characteristics of the vehicles from a set of training images that capture the allowed variability in the vehicle class. Usually, the variability of the non-vehicle class is also modelled in order to improve the performance. Also in these methods the previously introduced HG features are typically used. An example of a set of training images is shown in Figure 2.9.

Figure 2.9 - Examples of vehicle and non-vehicle images used for training [8].

Therefore, the output of the HV methods is an accurate “vehicle” or “non-vehicle” automatic classification.

2.1.3 Overview of some Vehicle Detection Methods

The main approach of the majority of the vehicle detection methods consists in following three basic steps:

1. Segment the moving objects from the frames of the video sequence (HG).
2. Extract some features of the segmented potential vehicle (HG/HV).
3. Classify the data and make a decision vehicle/non-vehicle (HV).

Most of the methods reported in the literature use a subset of the features presented in Section 2.1.1, combining knowledge-based with motion-based methods and then using template-based or appearance based approaches to define the position of a vehicle, if it is present in the image. A list of some of the most relevant published methods is presented subsequently:

- **Vehicle shadows and dimensions**

Knowing that in highways only motor vehicles are permitted, and with the goal of classifying vehicles passing in a toll into two classes, vehicles and non-vehicles, Shengli Shi et. all [10], proposed a simple detection method. After background subtraction, to segment the moving objects, Shengli considers shadows and the dimensions of the supposed vehicles, as the most important features to classify them. In order to detect shadows, vehicle colors are selected as a reference for the separation from gray or light black shadows. In the subtraction image, color parts are subtracted first, as vehicle candidate parts. A threshold can be automatically chosen to identify gray or black colors which represent shadows. After that, the algorithm focuses on vehicle height and width estimation using a single camera. A suit of sensors, consisting of an inductive loop detector and a piezoelectric loop detector, measure the length of the vehicle passing through the toll. This method has a relatively low computation cost, and it shows successful classification rates of approximately 92.2% for sensor detection and 90.4% for image segmentation.

- **Vehicle shadows, texture, edges and symmetry characteristics**

Using an initial segmentation based on shadow detection, Sheng Jin Li et. all, proposed a novel approach [7]. First, in order to obtain the gray information, this proposal selects an area in the underside and central part of the image as sample, this area is close to the CCD camera, so the acquired image includes less noise and can reflect better gray characteristics. As the gray value of vehicle bottom shadow area is lower than that of the whole road, the final threshold to segment the vehicle bottom shadow is described as:

$$Threshold = mean - k * Variance_{mean} \ (k \geq 1)$$

Here, *mean* represents the estimated mean value of the road surface, and *variance* represents the mean variance of the road surface, k is an adjusting coefficient, Sheng et. all show, with statistical results, that k=2 can segment the bottom shadow of vehicle very well, see example in Figure 2.10.

Figure 2.10 - Segmentation results of vehicle bottom.

Due to the problem of noise interference, it was found that the vehicle bottom shadow can be described as a rectangle while the noise does not have this shape. So, rectangularity of the shadow areas was adopted to eliminate noise areas.

However, some non-vehicles areas may also be considered because of variable illuminations and complicated road environments. Thus, Sheng Jin Li et. all chose three kinds of characteristics which are fused together to reduce the noise areas and judge whether the preceding object is a vehicle or not.

Knowing that there are many obvious texture characteristics in the area of the vehicle, like windshields, rear bumper and vehicle license plate, they estimate the roughness of the images, which is used to describe the image texture. Afterwards, knowing that there are obvious edges and symmetry characteristics in the rear of the vehicle, its vertical and horizontal edges as well as its symmetries are estimated.

In this way, according to Sheng Jin Li et. all the vehicles could be detected in the image since the left, right, bottom and upper edges and texture of the image are all acquired, and then integrated together.

The authors state that the algorithm is reliable and has good real-time performance, although no experimental results are shown to support this.

Based on the same features, Xuezhi Wen et. all took advantage of some aspects that Sheng Jin Li et. all had not considered and proposed a different method [8], as illustrated in Figure 2.11.

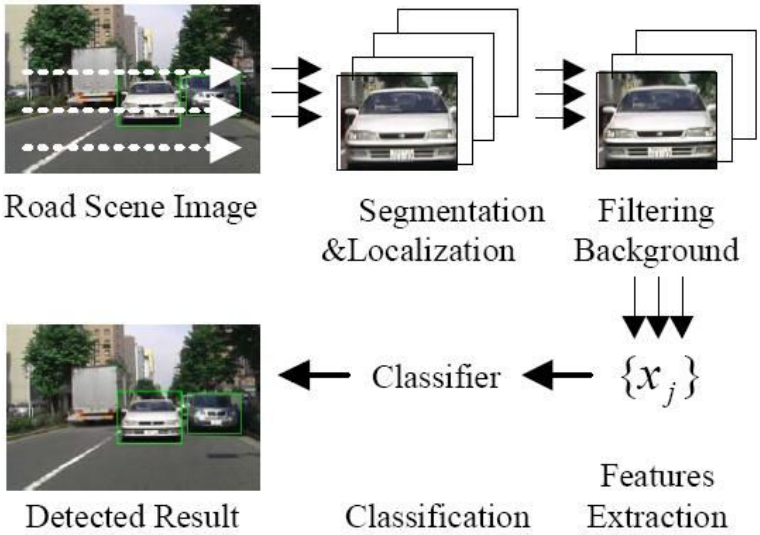


Figure 2.11 - Vehicle detection proposal by Xuezhi Wen et. all [8].

Instead of using a global threshold to detect the vehicle bottom shadow, they use the ratio between the neighbour pixels' gradient module value of a pixel and the grayscale value of that pixel to decide whether a pixel is a edge point of the shadow or not. Because the shadow underneath the vehicle is not always accurately localized at the vehicle bottom, depending on the sun (or other illumination source) position, the authors use the vehicle's vertical edges and symmetry to localize the vehicle accurately.

To calculate the vehicle's symmetry three methods were used: Binary Value Profile Symmetry [11], Grayscale Symmetry [12] and S Component Symmetry in HSV Space [12]. The final vehicle symmetry is synthesized by integrating the three components [13], which make the symmetry results' more accurate than the Sheng Jin Li et. all simple method's. Combining the three features, symmetry, edges (projection of the vertical edges) and shadow, a vehicle filter was build to detect potential vehicles position.

Finally, they construct an off-line vehicle and non-vehicle training set, based on 10933 non-vehicle images and 6488 vehicles images, in which a Wavelet Transform was performed for features extraction.

To evaluate the performance of the developed system, different road types were recorded, like highway (1200 images), urban narrow road (400 images) and urban common road (800 images). The method shows detection rates of around 90% for highways and around 88% for urban narrow roads and urban common roads.

- **Eigenspace**

Based on the eigenspace approach, and within the scope of developing a driver's assistance system that includes a traffic system monitoring, Anand Santhanam and M. Masudur Rahman proposed another moving vehicle detection technique [14]. This is an appearance based method consisting of two stages. Firstly, in the training stage the images were represented using PCA (Principal Component Analysis) [15], resulting in a low dimensional eigenspace, and thus a reduced computational cost. Secondly, in the testing stage, given an input image, the classification system projects it to the eigenspace, and then, instead of using a common distance classifier such as Euclidean distance to matching the testing sample with the eigenspace, the authors propose a new *"eigendimensions matching"* based on:

- Calculating the minimum and the maximum range of each eigendimension of the training datasets.
- Every selected eigendimension of the testing dataset should be larger than or equal to the minimum range of the corresponding eigendimension of the training dataset.
- Every selected eigendimension of the testing dataset should be smaller than or equal to the maximum range of the corresponding eigendimension of the training dataset.

Two different experiments were performed:

- Car-orientation classification (car-back, car-front and car-side);
- Non-car classification (images mostly include roads, pedestrians and road side objects).

Table 2.1 and Table 2.2 show a summary of experimental results and confirm the potential of the proposed method in comparison with some conventional methods, (Mean Eigenspace [14], Euclidean [35] and Mahalanobis [36] distances).

Table 2.1 - Non-car classification results.

Classifier used	Training Datasets	Testing Dataset	Classification Rate
Mean Eigenspace	4000	4000	95%
Eigendimension Matching	4000	4000	92%
Euclidean Distance	4000	4000	86%
Mahalanobis	4000	4000	51%

Table 2.2 - Car-orientation classification results.

	Car Orientations	Training Datasets	Testing Dataset	Classification Rate
MAHALANOBIS	*Car-back*	1800	1800	87.70%
	Car-front	90	90	49%
	Car-side	34	34	0%
	Car Orientations	**Training Datasets**	**Testing Dataset**	**Classification Rate**
MEAN EIGENSPACE	*Car-back*	1800	1800	92%
	Car-front	90	90	90%
	Car-side	34	34	88.2%
	Car Orientations	**Training Datasets**	**Testing Dataset**	**Classification Rate**
EIGENDIMENSION MATCHING	*Car-back*	1800	1800	94%
	Car-front	90	90	89%
	Car-side	34	34	88%
	Car Orientations	**Training Datasets**	**Testing Dataset**	**Classification Rate**
EUCLIDEAN DISTANCE	*Car-back*	1800	1800	56%
	Car-front	90	90	62.2%
	Car-side	34	34	76.5%

- **Edge points "attached with a descriptor"**

Another method presented in the literature was proposed by Xiaoxu Ma and W. Eric L. Grimson [16]. The proposed approach augments edge points as repeatable and discriminative features, combines several existing techniques with modifications to fit them better to the considered problem, and gives models that perform sufficiently well to serve the purposes. The feature extraction method is composed of:

- A tracking system [18, 19];
- Extracting edge points;
- Attaching a descriptor to each edge point;
- Segmenting edge points into points groups;
- Forming features from edge point segments.

According to the authors, repeatability of detected features is one of the pivotal factors for successful recognition, however it should be noted that there are still quite evident variations in edge images of objects from the same class, which makes a simple edge map-based technique ineffective. To solve this problem a SIFT descriptor (Scale Invariant Feature Transform) [17] was created by first computing the gradient magnitude and orientation at each image sample point in a region around an anchor point. The region is split into *r x r* sub-regions. An orientation histogram for each sub-region is then formed by accumulating samples within the sub-region, weighted by gradient magnitudes. Concatenating the histograms from sub-regions gives a SIFT vector, which was adopted with several

key modifications tuned to vehicle classification, as descriptors for edge points [16]. Afterwards, edge points that are both spatially close to each other and have similar descriptors can be grouped together. Edge point groups are advantageous compared with individual edge points. Firstly, edge point groups are more repeatable in terms of spatial locations. Secondly, edge point groups lead to more concise models. With edge points segmented, coordinates and associated SIFT vectors of edge points in one segment define a *feature*, as illustrated in Figure 2.12.

The most evident differences lie in the 9th to 12th elements in each descriptor, which correspond to the upper-right sub-region. The descriptors in the right column are useful to discriminate between the sedan and the taxi, since the upper-right sub-region of the taxi captures the textured area of the top light. Again the 9th to 12th elements of each descriptor are different. The collection of these features gives a good base to build models and classifiers. A changed *Constellation Model* [16] was build in order to create a probabilistic model to be used for classification. Then a Bayesian decision rule [16] gives the recognition result.

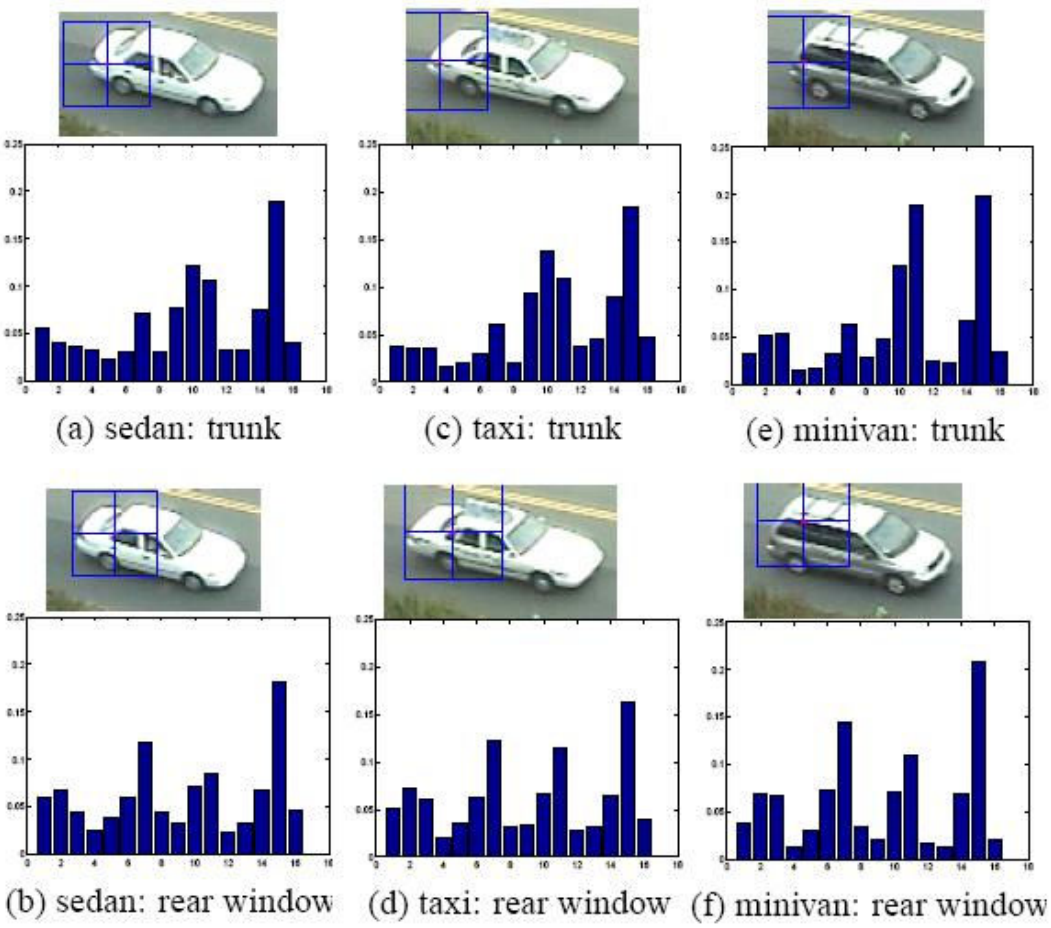


Figure 2.12 - Features examples [16].

Experimental results have two classification tasks: cars vs. minivans and sedans vs. taxis.

To build the models, for cars vs. minivans, 50 cars and 50 minivans were randomly selected from the dataset; for sedans vs. taxies, 50 sedans and 50 taxies. Another 200 sedans, 200 minivans and 130 taxies were selected for testing. All collected videos of traffic were obtained from an overlooking camera. Table 2.3 shows a summary of experimental results:

Table 2.3 - Experimental results for Xiaoxu Ma and W. Eric L. Grimson proposed method [16].

	Car	**Minivan**
Car	95%	5%
Minivan	5.5%	94.5%

	Sedan	**Taxi**
Sedan	92%	8%
Taxi	19.23%	80.77%

- **”U” Shape**

An interest approach was developed by Handmannet al. [20] who proposed a template based on the observation that the rear/frontal view of a vehicle has a “U” shape (i.e., one horizontal edge, two vertical edges, and two corners connecting the horizontal and vertical edges).

During verification, computer vision modules, generating lines (polygon approximation of the contour), local orientation coding, local image entropy and local variance analysis are coupled in a neural network. Later a fusion process is implemented for generating a saliency map interestingly similar as a “U” shape, as is illustrated in Figure 2.13. This template could be very fast, however, it introduces some uncertainties, because that there might be other objects on the road satisfying those constraints (e.g., distant buildings).

Figure 2.13 - "U" Shape *template-based* method [20].

2.2 Vehicle Recognition

As previously mentioned, not many existing solutions to recognize vehicle manufacturers and models are reported in the literature. However, the existing methods always use a combination of some of the features already presented in knowledge-based methods overview, to recognize the unknown vehicle.

Vehicle recognition can then be divided in two main parts:

1. **Feature Extraction** – Extracting discriminative features is the most important component of a vehicle recognition system, as those features need to capture each vehicle's unique and differentiating characteristics from the others. Most of the used features are similar to the ones described in Section 2.1.1.
2. **Feature Similarities** – Comparing the extracted features of the unknown vehicle with a database, allows to find which one looks the most like the unknown vehicle, and consequently to identify its manufacturer and model.

A list of some of the most relevant published methods found in literature is presented subsequently:

- **SIFT Matching**

In order to develop a robust vehicle surveillance system Louka Dlagnekov et all [24] decided to improve a license plate recognition method, with the recognition of vehicle make and model. Instead, of a normal vehicle segmentation, using a motion segmentation method, the authors use the location of the previously detected license plate as an indication of the presence and location of a vehicle in the video, and then, cropping a fixed window size around the license plate, extract the vehicle segmentation. This method can also be useful for make and model recognition in static images, where vehicle segmentation is a more difficult problem. In that way, the database set and test set were created and then ready to find the best match between them. Working upon the experiments developed by Lowe [29], Dlagnekov applied *Scale Invariant Feature Transform* (SIFT) to this problem of make and model recognition, mostly because SIFT is invariant to scale, rotation and even partially invariant to illumination differences.

The major stages of using SIFT to generate the set of image features [29] are:

1. **Scale-space extrema detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. **Keypoint localization:** At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

Knowing this, for each keypoint of a test image, Dlagnekov finds the keypoint of each database image that has the smallest distance to it and is at least a α factor smaller than the distance to the next closest descriptor. Counting the number of keypoints that are successfully matched between the test image and each database image, the database image with more keypoints successfully matched is considered the best match.

The SIFT matching algorithm described above achieves a recognition rate of 89.5% on the test set of 38 images and a database set of 1140 images including in some cases multiple images of the same make and model but different year in order to capture the variation of model design over time. An example of recognition results for some of the queries in the test set is shown in Figure 2.14.

Figure 2.14 - Example of recognition results, yellow lines indicate correspondences between matched SIFT features.

- **Edge map**

Trying to solve the vehicle make and model recognition problem and extending the work done by Dlagnekov [24], David Torres proposed an object classification technique based on the edges of the back end of vehicles [25]. Specifically, Torres generates a set of line segments that fit the back end of the vehicle's *edge map*, as illustrated in Figure 2.15. These line segments are then compared using a line segment Hausdorff distance developed by Gao and Leungin [26].

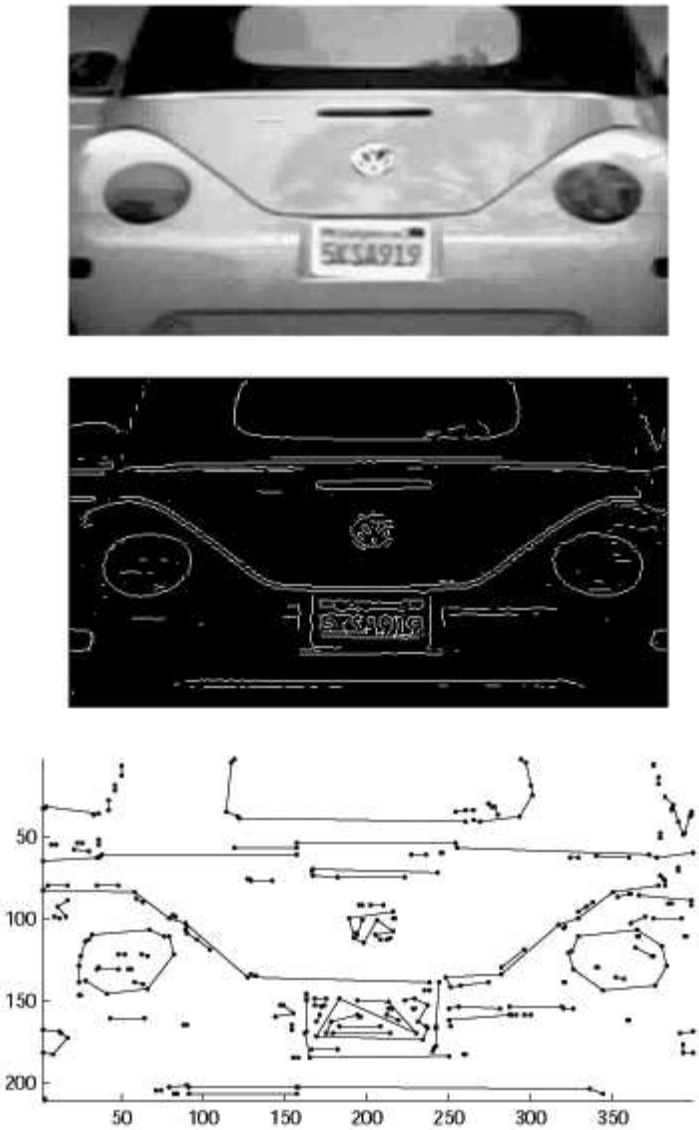


Figure 2.15 - An image, its edge map, and the generated line segments. [25]

To evaluate the performance of the developed system, Torres used the same database as Dlagnekov [24], which consist of 1103 cropped images of the back end of vehicles of various manufacturers and models. Also available is a set of 38 test images used to test the make and model recognition algorithm.

Table 2.4 shows a summary of experimental results. Note that the license plate portion of the images is cropped out to eliminate any noise that the region may contribute, and on experiment 2, in order to remove any noise which may occur due to the image background or to windshield reflection, the images are cropped, so that only the bumper portion of the image is visible, an example is shown in Figure 2.16.

Figure 2.16 - Sample cropped images used in experiment 2. [25]

Table 2.4 - Experimental results [25].

	experiment 1	**experiment 2**
Recognition Rate	0.4211	0.5966
Top 3 Rate	0.6579	0.7524
Top 5 Rate	0.7368	0.8264

- **Edge map mixed with redness measure**

Another method presented in the literature was proposed by Milos Stojmenovic [27], in order to build a real-time machine that could recognize Honda Accord 2004 from rear views. For the considered recognition problem a set of features, including a redness measure (Figure 2.17) and dominant edges in six orientation bins (Figure 2.18) were extracted.

Figure 2.17 - Example of redness detection result.

In addition to precision of detection, the second major aim was a real time performance, so applying a similar general design as in [28], developed by Viola and Jones in face recognition, Milos' program should quickly recognize all the vehicles of the given type and position in images within a second or so.

Figure 2.18 - Six orientations bins, Honda and corresponding edges.

At the end, based on a training set consisting of 155 "Honda Accord images" and 760 "non-Honda Accord images", with the feature set described above, back views of Honda Accords were detected with a 98.7% detection rate and 0.4% false detection rate on the training set. With a test set of 106 images that contain 101 Honda Accord in various scales, position and angle, their object recognizer performed with 89.1% detection rate and 26 false detections.

Chapter 3

Proposed Vehicle Recognition System

Following the discussion presented in Chapter 2, it is easy to conclude that an efficient system for vehicle manufacturer and model recognition is very dependent on the extraction of good features, as they are essential to distinguish one vehicle from another.

The vehicle manufacturer and model recognition method proposed in this Thesis only uses video sequences of vehicle rear views. This choice was based on the fact that distinguishing features can be extracted from a vehicle's rear view, such as symmetry, license plate and lights. On the other hand a system based only on vehicle rear views has less computational/implementation costs than another system that would require more than one view (rear, lateral or front) of a vehicle.

As discussed in the Introduction Chapter, there are several possible scenarios for a real application of the proposed system. Figure 3.1 illustrates some examples: (a) IST – Vehicle Database - *test set* scenario (Thesis database) - See Chapter 5 – Section 5.1; (b) Entrance of vehicle parking scenario; (c) Radar control scenario and (d) Toll control scenario.

(a) IST – Vehicle Database - *test set* scenario

(b) entrance of vehicle park scenario

(c) radar controls scenario

(d) toll controls scenario

Figure 3.1 - Examples of application scenarios for the proposed system.

In a real application the filming conditions should be correctly prepared for capturing images with the desired properties, for instance, by building a specific structure to hold the camera – see a possible example in Figure 3.2. This structure should be able to avoid some unwanted errors like reflections, wrong vehicle's position or overlapping vehicles, and so to improve the performance of the system. However, this was not possible to accomplish in the acquisition of the *IST* database test set, in order not to disturb the normal operation of the circulation within the campus.

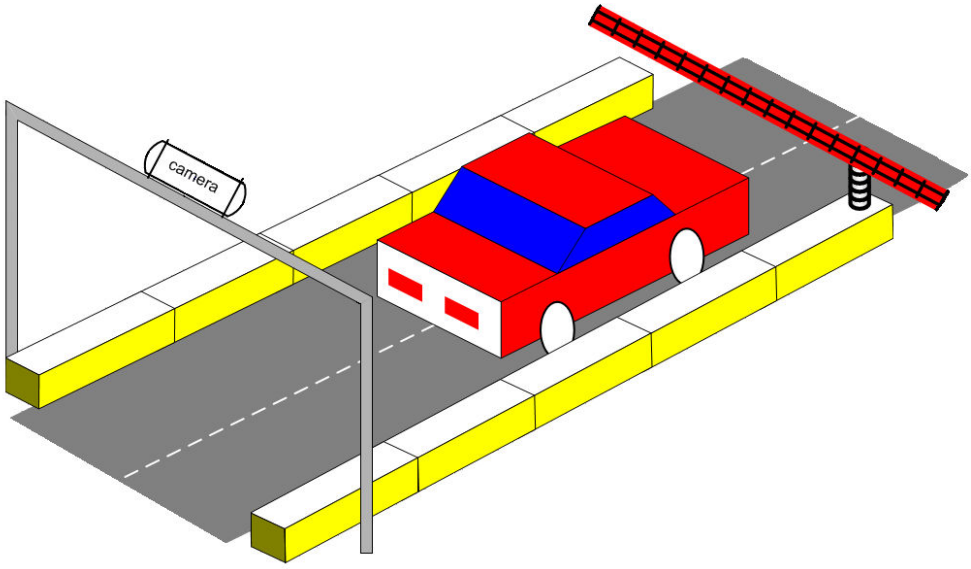


Figure 3.2 - A prototype to a proposed system structure.

The proposed vehicle detection and recognition system is composed by four main modules:

1. **Vehicle Segmentation**: The first module intends to correctly segment the “unknown” vehicles from the background and then extract, from the video, an image with an aligned rear view of each vehicle. This corresponds to the vehicle detection stage;
2. **Vehicle Feature Extraction**: The proposed *Vehicle Features Extraction* module has three main components, devoted to the computation of features to be used for recognition purposes. Three types of features are computed: *Vehicle Shape*, *Vehicle Lights and Vehicle Color*;
3. **Vehicles Database Processing**: Module number three is about the *System Database*, being used to add new vehicles to the database. It’s important to refer that to go on with Vehicle Recognition the *vehicles database* should be previously created – see section 3.3.
4. **Vehicle Recognition** (Manufacturer and Model): Finally, the fourth module, named *Vehicle Recognition*, computes the similarity between each “unknown” vehicle in a given video and all the rear view images of vehicles stored in the database, determining the one more likely to represent the “unknown” vehicle’s manufacturer and model. *Vehicle Color* is a feature given as output to characterize the “unknown” vehicle.

An overview of implemented method is presented, in the form of a block diagram, in Figure 3.3.

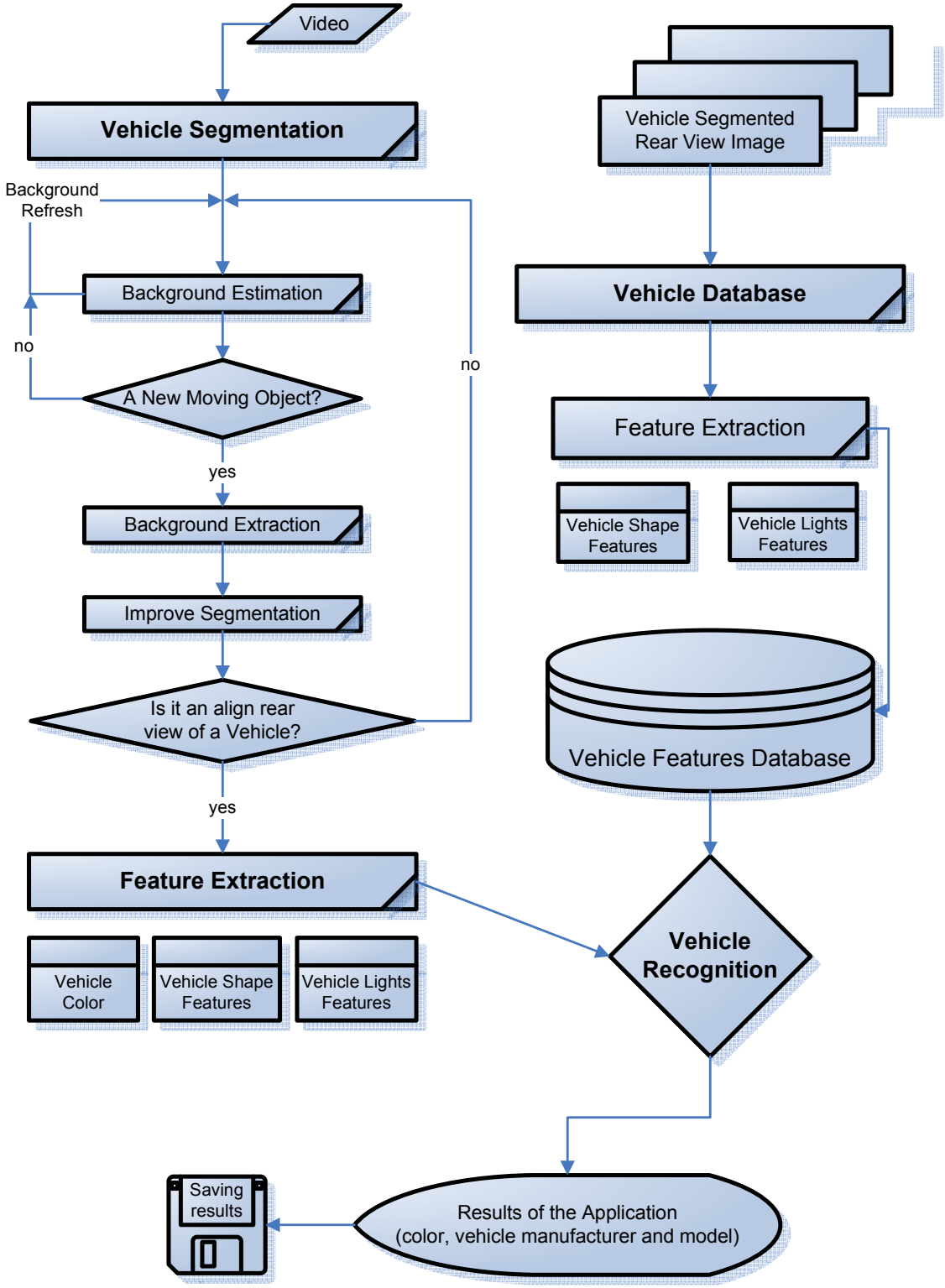


Figure 3.3 - Overview of the proposed Vehicle Detection and Recognition System.

The *Vehicle Segmentation* module is presented in Section 3.1. Then, the *Vehicle Feature Extraction* module is described in Section 3.2. Section 3.3 explains how the *Vehicle Database* is created and maintained. The final Section of this Chapter, 3.4, elaborates on the *Vehicle Color, Manufacturer and Model Recognition*.

3.1 Vehicle Segmentation

The first step for vehicle recognition based on the analysis of rear view image is to segment the video sequence into background and foreground, in order to correctly extract one aligned rear view image for each “unknown vehicle” – an example of an aligned rear view image is illustrated in Figure 3.1.

For this purpose a motion segmentation technique is employed, which relies on a correct estimation of the sequence background, and subsequent extraction. An overview of this procedure is shown in Figure 3.4 (n is the frame number of the video sequence):

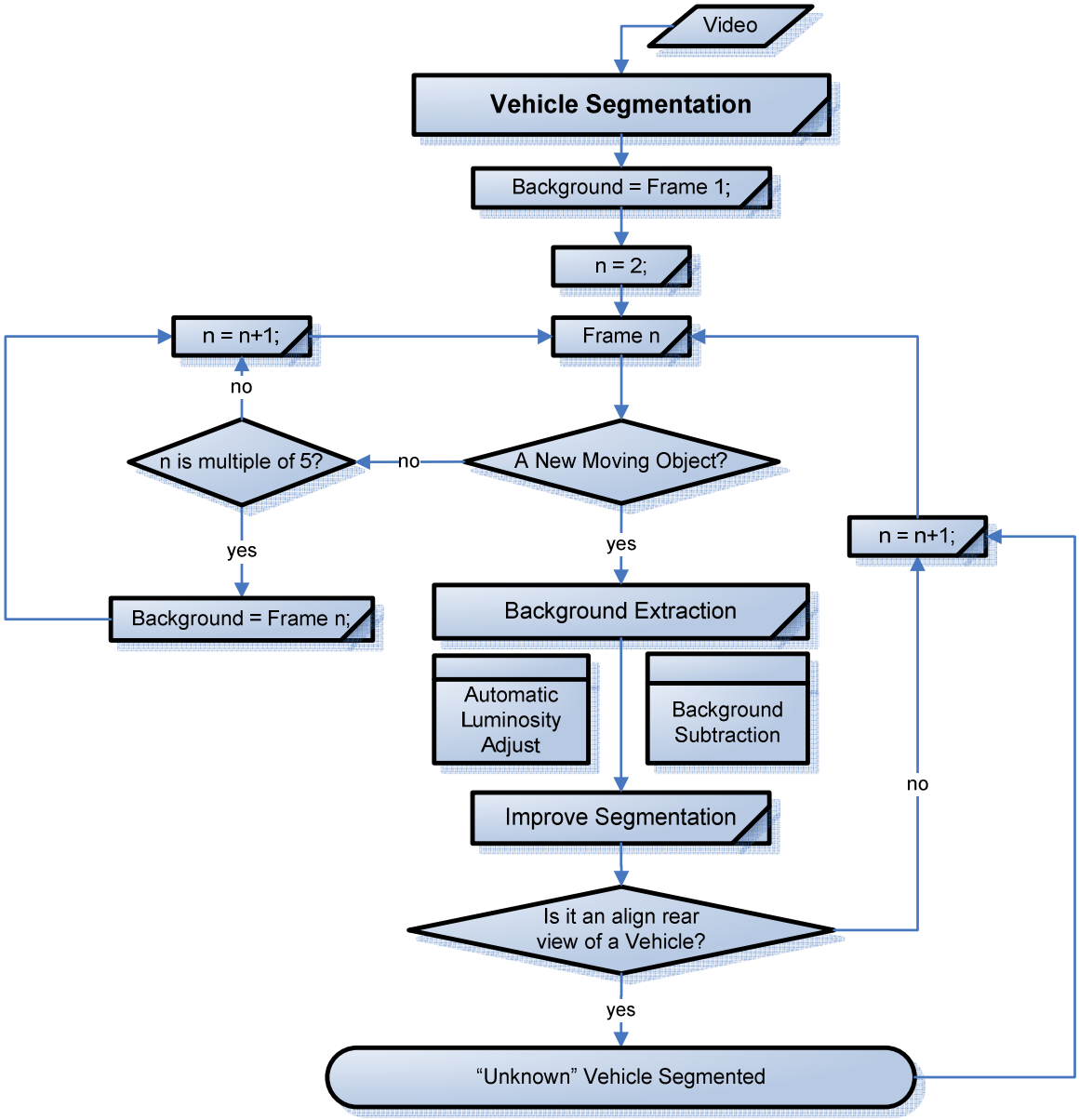


Figure 3.4 - Overview of vehicle segmentation module.

In the following, various steps are illustrated with images from a sample video captured for this Thesis, with a Renault Clio entering the parking garage of IST.

Background Estimation

In the type of application being considered, the camera capturing the video sequence is typically static. Therefore, the proposed system considers the first frame of a new sequence as a good background estimation. After that, a *background refreshment method* is implemented, which has the capability of, with an interval of 5 frames, updating the image of the background – see Figure 3.4. If a new vehicle appears in the video the method is able to only refresh the background image when the vehicle disappears. This allows the proposed method to become insensitive to luminosity changes, since the background image is continuously refreshed over time.

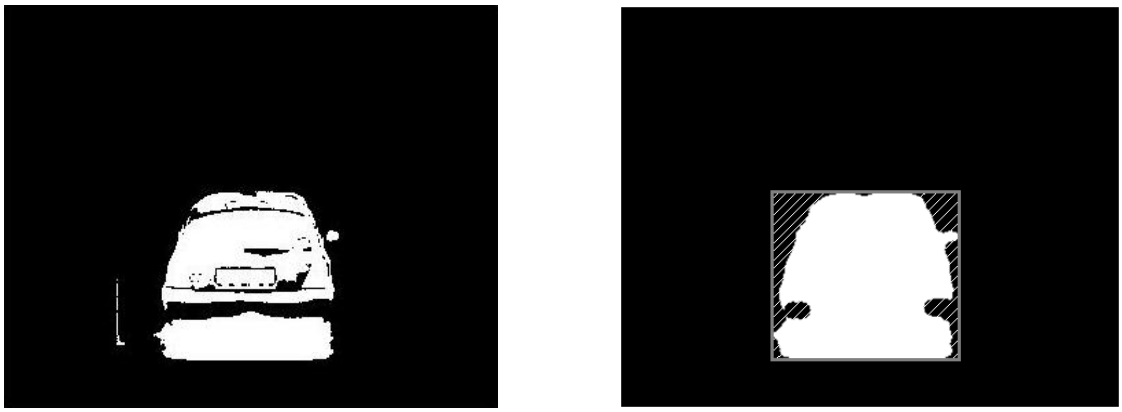
Background Extraction

One may suggest that simply performing background subtraction to the images of the video sequence is enough for foreground segmentation. In fact, this technique has been widely used to segment dynamic scenes in the cases where the camera is fixed. Thus, after detecting a new "unknown" object in the video and choosing a frame in which it is aligned with the camera, the proposed method performs an *automatic luminosity adjustment* between this frame and the updated background image currently available, before proceeding with the subtraction - an example is shown in Figure 3.5.

Figure 3.5 - An example of a background subtraction of an aligned vehicle rear view.

Segmentation Improvement

It's easy to understand that some of the pixels in the subtraction image belong to the background. These are considered to be noise pixels due to camera noise, slight illumination variations, perturbations in the camera position, shadows, etc. In order to improve the binary segmentation of the vehicle it is necessary to find a threshold that rejects most noise pixels. To achieve a low computation cost, a global threshold using Otsu's method [30] is selected. However, the resulting binary image, see Figure 3.6, typically still contains noise pixels and errors that need to be further processed.



Otsu's method result Main binary region

Figure 3.6 - An example of a binarization based on Otsu's method [30].

To improve the segmentation result an adaptable threshold for different parts of the vehicle is applied [31]. The process starts by isolating the region of the grayscale difference image which contains the moving vehicle. This is done by masking that image with a segment that contains the *main binary region* obtained above – see result in Figure 3.7. At this stage, base on the characteristics of this *region* (area, symmetry), the system can decide if the "unknown" object is a vehicle or not.

Figure 3.7 - Example of a masked difference image.

The segmented vehicle image is now divided into 2 pixels' height blocks and then Otsu's method is applied to each block. This way, it is believed that the vehicle segmentation can be enhanced because a suitable local threshold is chosen for a small region rather than for the whole vehicle's shape, avoiding to miss out important parts, or to add considerable amounts of noisy pixels. An example of several steps of this process is included in Figure 3.8.

Figure 3.8 - Example of the progression of an adapted binarization and final segmentation.

Finally, the image is two-dimensionally median filtered, and a binary dilation and erosion is performed in order to enhance the vehicle's shape contours, as illustrated in Figure 3.8.

3.2 Vehicle Feature Extraction

As mentioned above, the proposed method uses all the features that can be correctly extracted from an aligned rear view of a vehicle to proceed with its recognition. The *Vehicle Feature Extraction* module is composed by three main parts:

- Vehicle Shape Feature Extraction;
- Vehicle Lights Feature Extraction;
- Vehicle Color Feature Extraction.

3.2.1 Vehicle Shape Feature Extraction

Looking at the rear image of a vehicle it is easy to understand that there are several features that can be extracted from its shape, to be used for recognition purposes. The proposed system is implemented to extract three such vehicle's features, notably: a *width/height coefficient*, a *binary edges map and* the *vehicle's outer contour*.

The vehicle shape features are extracted using the *Vehicle Segmentation* results, explained in the previous Section and illustrated in Figure 3.9 (a). However this segmentation is further improved with a "*horizontal mirroring method*", developed by the author. Based on some horizontal symmetry points in the vehicle shape, this method allows the system to fix some remaining errors. Figure 3.9 (b) shows an example of the result obtained using this technique.

Considering that the bottom part of almost all vehicles' shapes (rear bumper, wheels, etc) is very similar, only the top ¾ of the vehicle segmentation is used for shape features extraction, to highlight each vehicle's singularities - see Figure 3.9 (c).

Figure 3.9 - Vehicle shape's enhancement.

In the following, each of the vehicle's shape features used for recognition is briefly presented.

- **Width/height coefficient**

To be robust to variable distances to the static camera, all vehicles' rear images were initially resized. However, the vehicle's original shape aspect ratio (width/height coefficient) was maintained and can provide a good filter to distinguish some vehicles. Therefore, before each segmented vehicle is cropped into ¾ of its height – see Figure 3.9 (b), its *width/height coefficient* is computed – see some examples in Table 3.1 (b).

- **Binary edge map**

It is easy to understand, while looking into a vehicle from a rear position, that its edges are among the most distinguishing visual characteristics that people use to differentiate vehicles between themselves. Having this consideration well present, the relevance of using the vehicles' rear view edges as a feature in the proposed method, is understandable. Therefore a "classic" Sobel Filter is applied to obtain an Edge map of each vehicle shape, which is then improved with a 2x2 dilatation, as the example in Figure 3.10 illustrates.

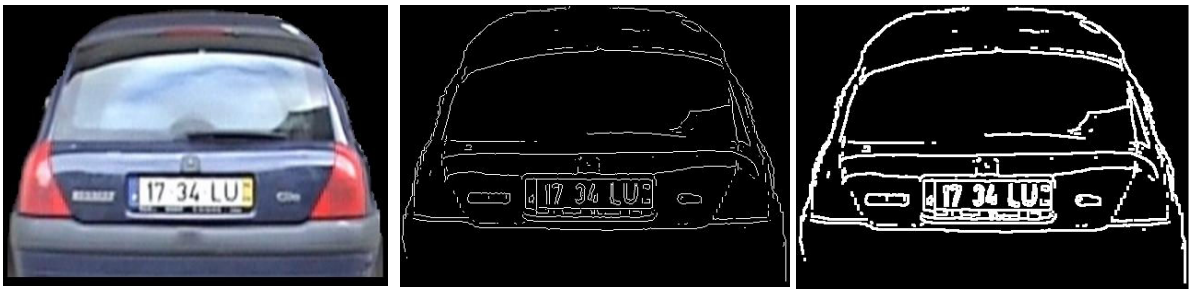


Figure 3.10 - Example of a vehicle rear view binary edge map.

Some more examples of binary edge maps are included in Table 3.1 (d).

Based on the horizontal symmetry of a vehicle's rear view, the binary edge map will be divided into two different parts that will be independently used during the recognition process - see Section 3.4.1. This option allows the proposed recognition algorithm to become more robust to some unwanted errors caused e.g., by shadows, foreign objects or some unexpected changes in the vehicle position.

- **Vehicle's Outer Contour**

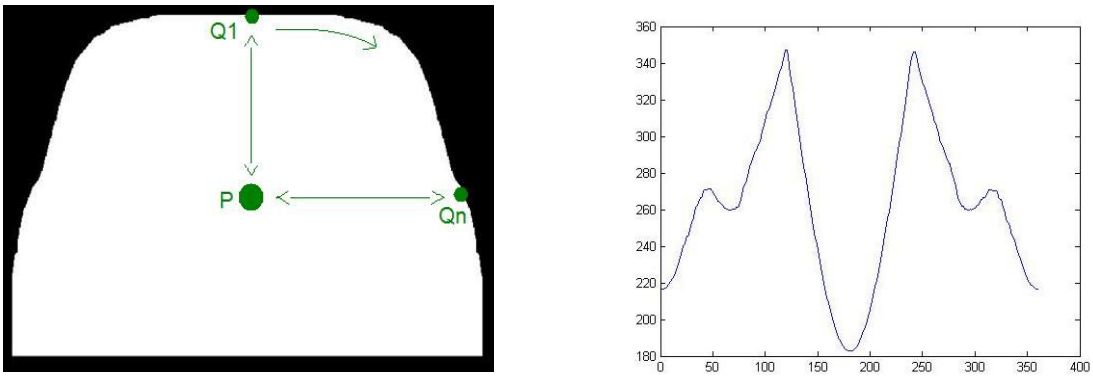
The third feature considered in this proposal represents a novel approach in the field of vehicle recognition. It has been based on the work developed by several authors e.g., for human gait recognition [31, 33]. This proposal consists in tracing the outline of the binary vehicle's shape, i.e, its outer contour, following its border points in an ordered way, to allow the computation of a signal with the distances between these contour pixels and the shape's centroid. Therefore, the centroid of the vehicle's shape is first determined, and the top-most point of the contour of the shape is selected as being the last white pixel going upwards from the centroid. This point is chosen as the starting point to perform the shape outer contour following.

The Euclidean distances (1) between the coordinates of each outer contour pixel, $Q = (r,s)$, and the centroid, $P = (p,q)$, are computed and stored, forming a distance signal, which can be represented as illustrated in Figure 3.11.

$$|PQ| = \sqrt{(p-r)^2 + (q-s)^2} \tag{1}$$

The distance signal is then normalized in terms of length, i.e. the number of points composing the vehicle's shape contour, in order that all distance signals have the same dimensionality, and can be more easily matched. The normalized distance signals have a fixed length of 360 points, corresponding to 360 angles with variations of 1º around vehicle's shape contour.

The signal is also 1-dimensionally median filtered using a 5 points structuring element, so that minor flaws in the shapes can be overcome. Table 3.1 (f) shows some examples with two different manufacturer and model vehicles.

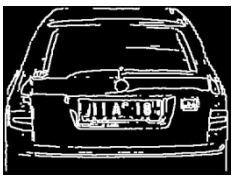
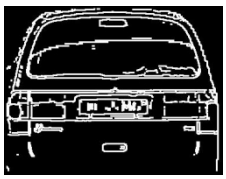
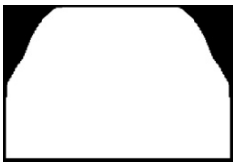
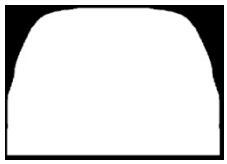
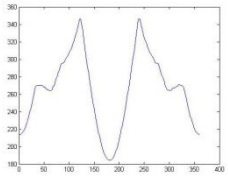
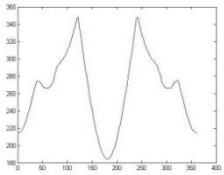
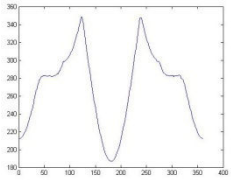
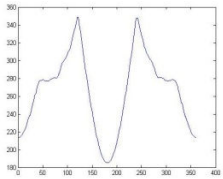


Vehicle's binary shape

Normalized Distance Signal

Figure 3.11 - Distance Signals' samples.

Table 3.1 - Two different vehicles (test set and database set) and their extracted shape and respective features values.

	test set	*database set*	*test set*	*database set*
Segmented Vehicle				
Improved Vehicle Shape (a)				
Width/height coefficient (b)	1.15	1.16	1.08	1.10
¾ of the Improved Vehicle Shape (c)				
Binary edge map (d)				
¾ of the initial binary shape (e)				
Vehicle's Outer Contour (Distance Signal) (f)				

Even though the features between different vehicles with same manufacturer and model are not identical, it is easy to conclude that, comparing them with the same features of another vehicle with different manufacturer and model, they present substantial differences and prove to be useful for a vehicle recognition system.

Section 3.4.1 details the way the proposed method uses these features to compute the similarities between each vehicle from the test and database sets.

3.2.2 Vehicle Lights Feature Extraction

Based on what has been written in Chapter 2, Section 2.1.1, it's easy to conclude that one of the most singular features in each different vehicle is its pair of back lights. Looking at Figure 2.5 is understandable that its position, orientation, symmetry, color and shape, typically make each vehicle quite unique.

In fact, when we take a look at a rear view of a vehicle, their lights combined with its shape are the main cues that we unconsciously use to recognize the vehicle manufacturer and model. Knowing this it's necessary to develop a robust and reliable vehicle lights detection and extraction method.

Generally, most vehicle lights have red as their main and more relevant color. Therefore, the first step of the detector should be to correctly find all the red color segments appearing in the segmented vehicle's image.

The method proposed to detect the red back lights of vehicles is based on the algorithm proposed by Carlos Paulo in [34], and it had never been used for vehicle recognition purposes, thus including some adjustments developed by this Thesis author.

Instead of taking advantage of a typical HSV conversion, a new conversion based on the HSV has been adopted, as described in the following.

For an input image in RGB format, the typical conversion to HSV (Figure 3.16) is done according to equations (2), (3) and (4), where *MAX* and *MIN* are equal to the maximum and minimum of (R, G, B) pixel values, respectively.

$$H = \begin{cases} undefined, & MAX = MIN \\ 60 \times \dfrac{G - B}{MAX - MIN}, & (MAX = R) \wedge (G \geq B) \\ 60 \times \dfrac{G - B}{MAX - MIN} + 360, & (MAX = R) \wedge (G < B) \\ 60 \times \dfrac{B - R}{MAX - MIN} + 120, & MAX = G \\ 60 \times \dfrac{R - G}{MAX - MIN} + 240, & MAX = B \end{cases} \tag{2}$$

$$S = \begin{cases} 0, & MAX = 0 \\ 1 - MAX, & otherwise \end{cases} \tag{3}$$

$$V = MAX \tag{4}$$

Since for vehicle lights detection only the red color is of interest and to avoid pixels where the red color is not well defined, the author proposes the usage of a modified hue/saturation–based detection functions for red, according to equations (5) and (6), adopted from [34]:

$$hd_{red} = \begin{cases} 1 - \dfrac{|G - B|}{MAX - MIN}, & (MAX = R) \wedge (MAX - MIN \geq th) \\ 0, & otherwise \end{cases} \tag{5}$$

$$S = \begin{cases} 0, & MAX - MIN < TR) \\ 1 - MAX, & otherwise \end{cases} \tag{6}$$

This function (5), as before, gives the red color probability, with values ranging from 0 to 1, where a higher value corresponds to a higher color probability. On the other hand, the saturation value (6) is estimated by setting to 0 all values where the difference between *MAX* and *MIN* is below a threshold value, *TR.* A threshold value of 0.1 [34], sets aside the areas that presented good saturation values but for which the color was not well defined.

This strategy not only reduces the computation time, but also has great results in red color detection as shown in Figure 3.12.

Figure 3.12 - Vehicle Lights detection.

In order to extract some light parameters that will be used in vehicle recognition it's necessary to obtain its binary image.

Like in vehicle shape, the Otsu's method [30] is selected to binarize the images. Finally, to improve the segmentation result, a binary dilation and erosion is performed. An important element of the proposed method is that, for each pair of vehicle lights (symmetric by design), the system chooses the one that contains a higher number of white pixels to represent the back light shape, to reduce the effects of noise or possible partial occlusions. This is illustrated in Figure 3.13.

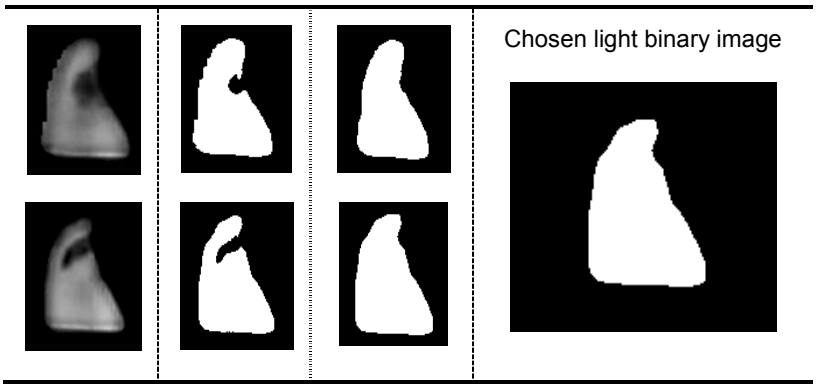


Figure 3.13 - Vehicle light binarization.

The vehicle back lights can now be characterized by several features computed from their shape, as well as to be used to compute additional geometrical features to be used for vehicle recognition purposes. These features are described in the following.

- **Eccentricity**

Defined as the ratio of the distance between the focus of the ellipse and its major axis length, *eccentricity* is a good mathematical parameter to characterize 2-D circular objects and its shape properties. *Eccentricity* value is normalized between 0 and 1, where an object whose eccentricity is 0 is actually a circle, while an object whose eccentricity is 1 is a line segment. Table 3.2 (a) includes examples of eccentricity values computed for four different manufacturer and model vehicles (test set and database set) segmented back lights.

- **Orientation**

Another distinctive feature between vehicle lights is its orientation, which, in the proposed work, is defined as the angle between the x-axis and the major axis of the ellipse that has the same second-moments as the region, as illustrated in Figure 3.14.

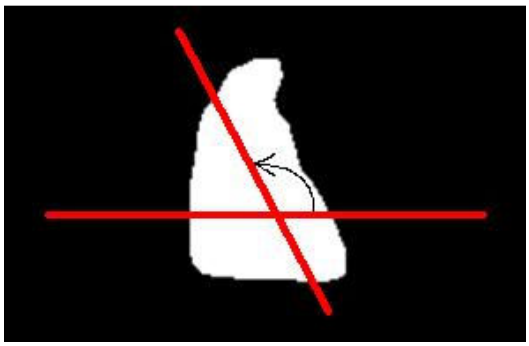


Figure 3.14 - Vehicle Light orientation angle.

The importance of this feature is demonstrated in Table 3.2 (b).

- **Position**

Looking at the rear of a vehicle it's easy to understand that back lights may have different "y-axis" positions depending on manufacturer and model. Thus, assuming a position to have value "0" when the center of the light is at the top of the vehicle, and "400" when it is at the bottom, a new reliable measure for vehicle recognition is proposed. Some examples of the position feature values are included in Table 3.2 (c).

- **License Plate Angle**

Knowing that all vehicles have a license plate and its position is variable depending with the vehicle manufacturer and model, another new feature is proposed. *License plate angle* is the angle between the x-axis that corresponds to the bottom of the license plate and a line that crosses the center of the back light on the left and the bottom of the license plate in the center of the vehicle, as illustrated in Figure 3.15.

Figure 3.15 - Angle between vehicle light and license plate.

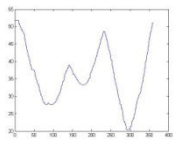
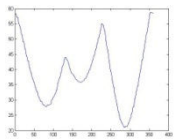
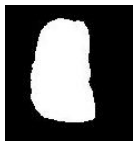
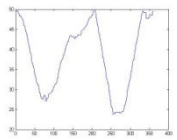
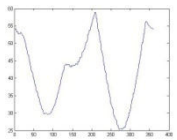
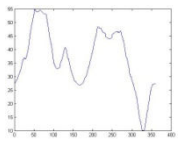
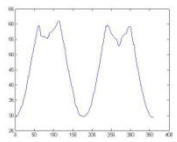
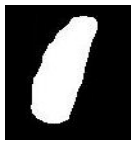
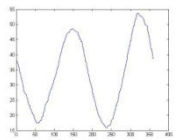
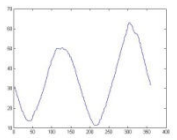
In order to correctly estimate this feature it is necessary to develop a license plate detection method. Based on HSV color space, see Figure 3.16, the proposed method aims to find the yellow segment in the license plate (vehicle year and month) in order to find its bottom position. This could be a flawless detection method, knowing that license plates are standardized for each country, and the vast majority of vehicles in the European Union have this specification. However it is difficult to find license plates that have exactly the same original color and this difference tends to grow for older license plates where the color has changed due to environmental conditions like sun exposure. Although, when correctly detected, this is a powerful feature as demonstrated in Table 3.2 (d). It is Important to mention that if this feature is not correctly estimated, or the vehicle main color is yellow, it will not be considered for the recognition step.

- **Light's Outer Contour**

Similarly to what was proposed to describe the *vehicle's outer contour* (Section 3.2.1), a *distance signal* is computed, now to characterize each back *light's outer contour*.

Some examples of the obtained light distance signals are shown in Table 3.2 (e).

Table 3.2 - Four different vehicles (test set and database set) and their extracted light and respective features values.

	vehicle	*Extracted light*	*Eccentricity (a)*	*Orientation (b)*	*Position (c)*	*License Plate Angle (d)*	*Light's Outer Contour (Distance Signal) (f)*
test set			0.7638	107.4º	201.7	7.3º	
database set			0.78	101.9º	197.3	5.77º	
test set			0.82	91.76	208.5	2.8º	
database set			0.83	98.1	192.7	2.84º	
test set			0.8329	9.3	177.2	8.7º	
database set			0.84	0.9	184.4	6.6º	
test set			0.93	72.5º	74.7	60.8º	
database set			0.97	64º	68.6	52.2º	

Looking at Table 3.2, it is easy to conclude that, with this group of light features, each vehicle is unique and can be distinguished from the others. However, when the vehicle's main color is red, none of these features can be used and so, the recognition will have to rely in the vehicle's shape features.

3.2.3 Vehicle Color Feature Extraction

In security and surveillance issues it is understandable that indentifying the vehicle color is an important task, and as such it must be considered by the system being proposed. Unlike what can happen with the vehicle's license plate, the color is something that is hard to change in a short time and therefore constitutes a powerful recognition feature. On the other hand the vehicle's color is something that can be easily identified at a considerable distance which can prove to be very useful when the goal is, for example, vehicle recognition in a police persecution, or some other recognition task that cannot be performed at a close distance.

To perform the vehicle's color detection, the proposed method uses a well-known color model, formally first described in 1978 by Alvy Ray Smith, called HSV [32]. HSV is a representation which attempts to describe perceptual color relationships more accurately than RGB, while remaining computationally simple. HSV describes colors as points in a cone whose central axis ranges from black at the bottom to white at the top with neutral colors in between. The angle around the axis corresponds to "**H**ue", the distance from the axis corresponds to "**S**aturation", having fully-saturated colors around a circle at the top, and distance along the axis corresponds to "gray **V**alue", Figure 3.16.

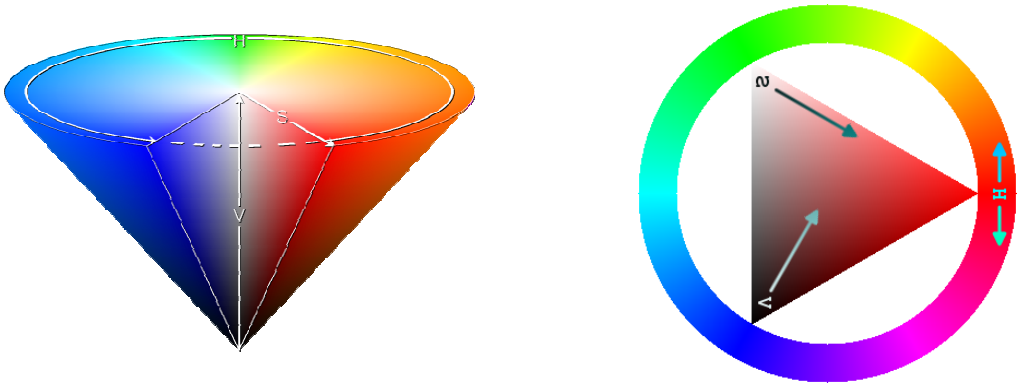


Figure 3.16 - Representation of colors in the HSV system. Taken from en.wikipedia.org [32].

To improve the color detector, only the middle part of the vehicle's rear view is considered, avoiding some elements like windshields and rear bumper that usually don't have the same color as the rest of the vehicle. After that, and based on the characteristics of HSV color space, the vehicle´s

color is easily detected. An example of several steps of this process is included in Figure 3.17.

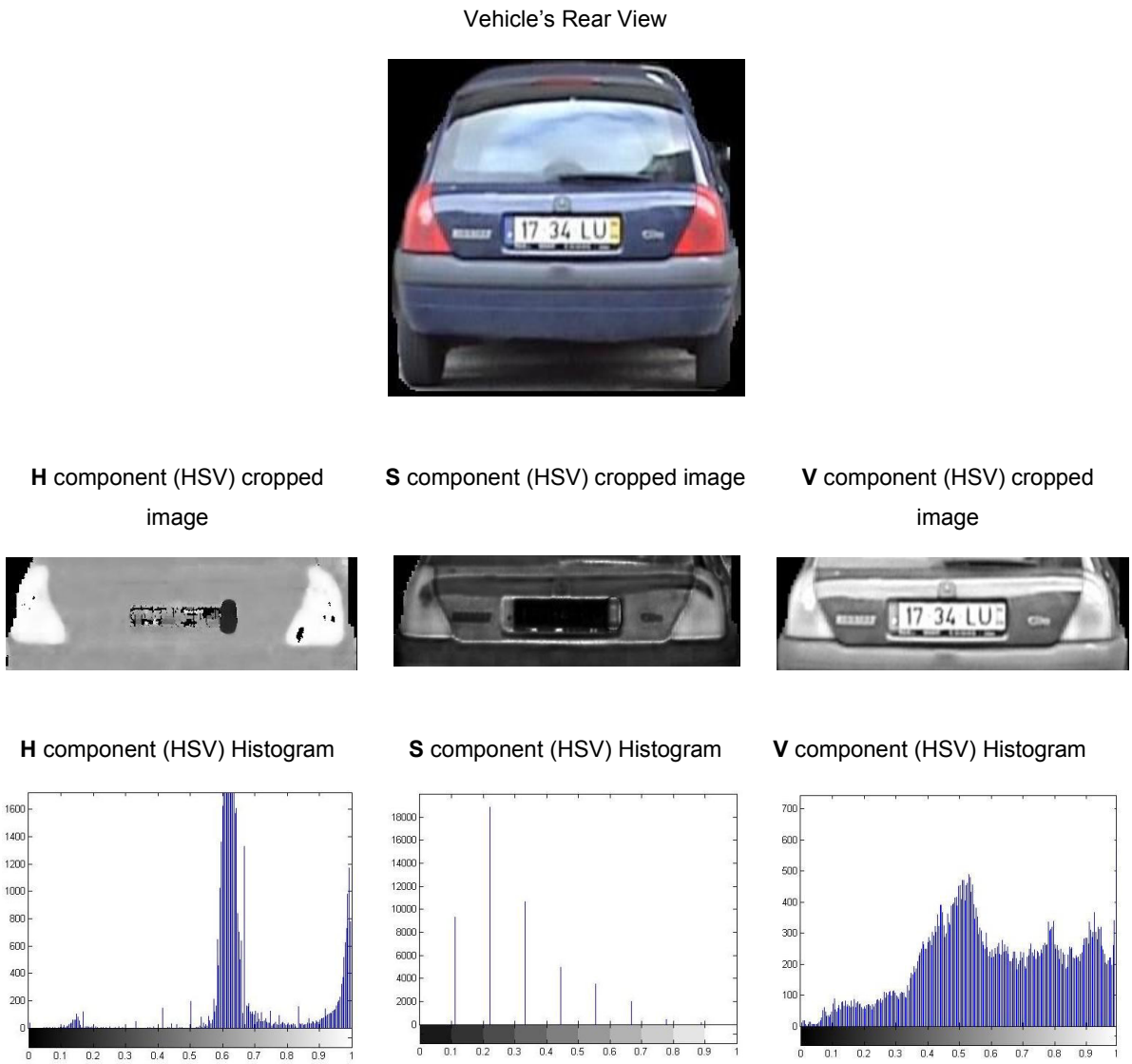


Figure 3.17 - An example of several steps of color detection process.

Most of the pixels in the cropped image have the color of the vehicle, thus, by plotting the correspondent HSV components histograms, it is possible to choose the Hue, Saturation and gray Values, that characterize the "unknown" color and then, detect it. In this particular case, with these three histograms and looking at Figure 3.16, it's easy to conclude that, being **H**ue's angle around 220º, **S**aturation approximately 3 (scale 1-10), and gray **V**alue, which decide if a color is dark or light, about 50%, the detected color is *"blue"*.

Based on the fact that sometimes it's not easy to distinguish one color from another, ten color categories were implemented: *white or light gray*, *black or dark gray*, *gray*, *red*, *orange*, *yellow green*, *blue*, *purple* and *Pink*.

3.3 Vehicle Database Processing

In any recognition system a good *Database* is essential, so the same happens in a Vehicle Recognition System. The database has a major relevance since it is one of the responsible for a high recognition performance.

As shown in the System Overview Diagram - see Figure 3.3, *Vehicle Database* includes the module "*Vehicle Feature Extraction"* from the main recognition system, however, with a slight difference, as in this case *Vehicle Feature Extraction* is not applied to vehicles segmented from an input video, but to rear view vehicle images previously loaded to the system. In order to keep the system up to date, its database should be continuously updated with new manufacturer and model vehicles.

Vehicle Database specifications and all the procedure to correctly add a new vehicle to it are presented in Chapter 4, section 4.2.2.

Some examples of features extracted from a vehicles database are shown in Table 3.1 and Table 3.2.

3.4 Vehicle Recognition

Based on what was written above, it is easy to conclude that *Vehicle Manufacturer and Model Recognition* is performed through a similarity computation between each "unknown" vehicle in a given video and all the rear view images of vehicles stored in the database.

The computation of similarities is done separately for the features computed from the vehicle shape and from the back lights:

- ***Shape Similarity:*** Similarity between all the features extracted from vehicle shapes;
- ***Lights Similarity:*** Similarity between all the features extracted from vehicle back lights.

After computing these two similarity values, they are combined (*Vehicle Similarity*) for recognizing the database vehicle more likely to correspond to the "unknown" vehicle being tested, thus identifying its manufacturer and model. The vehicle's color is also automatically determined, as explained in Section 3.2.3.

As mentioned before, when the vehicle's main color is red the features computed form the back lights cannot be used and so only shape similarity is used for recognition purposes.

A brief explanation of each type of similarity computation is presented in Sections 3.4.1 and 3.4.2.

3.4.1 Shape Similarity

To perform *Shape Similarity* it is necessary to compute the similarity between each extracted shape feature of a given probe (i.e. "unknown" segmented vehicle) and all the vehicles with descriptions stored in the database:

1. Width/Height Coefficient Similarity (***WHCS***): Since this coefficient is a number, its similarity is obtained by a simple normalized subtraction (7):

$$WHCS = \frac{|WHC_{probe} - WHC_{database}|}{MAX(WHC_{probe}, WHC_{database})} \tag{7}$$

2. ¾ of Vehicle Binary Segmentation Similarity (***VBSS***): To compute the similarity between Vehicle Binary Segmentations, a correlation method is implemented (8), where *VBS* are matrices of the same size [*m n*], each one corresponding to a Vehicle Binary Segmentation – see examples in Table 3.1 (e).

$$VBSS = \frac{\sum_{m,n}(VBS_{probe}.VBS_{database})}{\sum_{m,n} VBS_{probe} + \sum_{m,n} VBS_{database}} \tag{8}$$

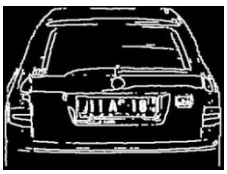
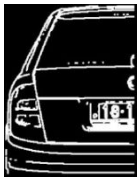
3. Binary Edge Map Similarity (***BEMS***): In order to estimate the *Binary Edge Map similarity* between probe and database Maps, a Normalized 2-D cross-correlation is implemented. The implementation follows the procedure proposed by J. P. Lewis [37]:

   - Calculate cross-correlation in the spatial domain.
   - Calculate local sums by pre-computing running sums.
   - Use local sums to normalize the cross-correlation to get correlation coefficients.

   Thus, the higher correlation coefficient is considered as the similarity value between probe and database Maps.

   Like has been written above in Chapter 3, section 3.2.1, the binary edge map is divided in two different parts that will be independently used on recognition process. Table 3.3 illustrates this process.

Table 3.3 - Example of *Binary Edge Maps Similarity* calculation.

¾ of the Improved Vehicle Shape		
Binary edge map		
Binary edge map division	  A1 A2	  B1 B2
Normalized 2-D cross-correlation maximum values	(A1,B1) = 0.20 (A1,B2) = 0.18 (A2,B1) = 0.16 (A2,B2) = 0.15	
Selected Binary Edge Map		
Binary Edge Map Similarity	*BEMS* = 0.20	

4. <u>Vehicle's external shape Distance Signal Similarity (***VDSS***)</u>: Euclidean distance between the computed distance signals for the probe and the database vehicles is computed in order to estimate their similarity (9):

$$VDSS = \sqrt{\sum_{i=1}^{n} (VDS_{probe\ i} - VDS_{database\ i})^{2}} \tag{9}$$

Where *VDS* are vectors of the same size [*n*], each one corresponding to a *Vehicle's external shape distance signal* – see example in Figure 3.11.

Finally, the *Shape Similarity* is obtained according to equation (10), in which different "weights" are given to the extracted feature similarities.

$$Shape\ Similarity = 10 \times (1 - WHCS) + 20 \times VBSS + 30 \times BEMS + 40 \times (1 - VDSS) \quad (10)$$

These different "weights" were selected from an exhaustive set of tests to give more importance (larger weights) to the more discriminative features.

3.4.2 Lights Similarity

Once again, to calculate the *Lights Similarity*, is necessary to compute the similarity between each extracted light features of a given probe (i.e. "unknown" segmented vehicle) and stored vehicles in the database:

1. Lights Binary Shape Similarity (***LBSS***): This similarity is computed using the same correlation method explained for the *shape similarity*, using equation (8). Table 3.2 shows some examples of lights binary segmentations.

2. Eccentricity Similarity (***ES***) / Orientation Similarity (***OS***) / Lights Position Similarity (***LPS***) / License Plate Position Similarity (***LPPS***): Since these features are single numbers their similarity is obtained by a simple normalized subtraction, using equations similar to (7).

3. Light's external shape distance signal Similarity (***LDSS***): This similarity is computed using the same method explained for the shape similarity, according to equation (9).

Combining all the individual feature contributions, the Lights Similarity can be computed by (11):

$$Lights\ Similarity = 21 \times LBSS + 15 \times (1 - ES) + 15 \times (1 - OS) + 15 \times (1 - LPS) + 16 \times (1 - LPPS) + 18 \times (1 - LDSS) \quad (11)$$

Once again these "weights" were arranged to give more importance to the more discriminative features.

3.4.3 Vehicle Similarity

Once *Shape and Lights Similarities* have been computed, the final *Vehicle Similarity* is calculated according to (12):

$$Vehicle\ Similarity = 40 \times Shape\ Similarity + 60 \times Lights\ Similarity \quad (12)$$

These "weights" were estimated knowing that *Lights Features* are less dependent of the vehicle's segmentation quality, and so are expected to be more robust than the *Shape Features*.

This way, the system is able to determinate the database vehicle more likely to be the "unknown" vehicle, and so, to conclude about its manufacturer and model, Figure 4.4 shows an example of system output.

Chapter 4

Software Implementation and the Graphical User Interface

This Chapter provides a brief explanation of the options taken for the software implementation and the choice of a development environment. The graphical user interface (GUI) is also described, highlighting the most important aspects using the implemented software.

4.1 Development Environment

To establish an automatic vehicle recognition system, a robust programming language that could handle images with relative ease, if not specifically oriented to image processing, is required.

To implement the aforementioned methods, several types of programming languages could be used. Taking into account that a major part of this project focuses on the manipulation of matrices and vectors, a programming environment able to handle these data types efficiently has been selected. Therefore, the choice was to use *Matlab* and its *Image Processing Toolbox*.

Matlab has the advantage of providing many image and matrix handling functions to be readily used, but on the other hand when it comes to simple programming functions it can become quite slow and/or complex (e.g. cycles/loops such as *for* or *while* functions).

4.2 Graphical User Interface

When the vehicle recognition program is started, the user is presented with an initial window, as illustrated in Figure 4.1.

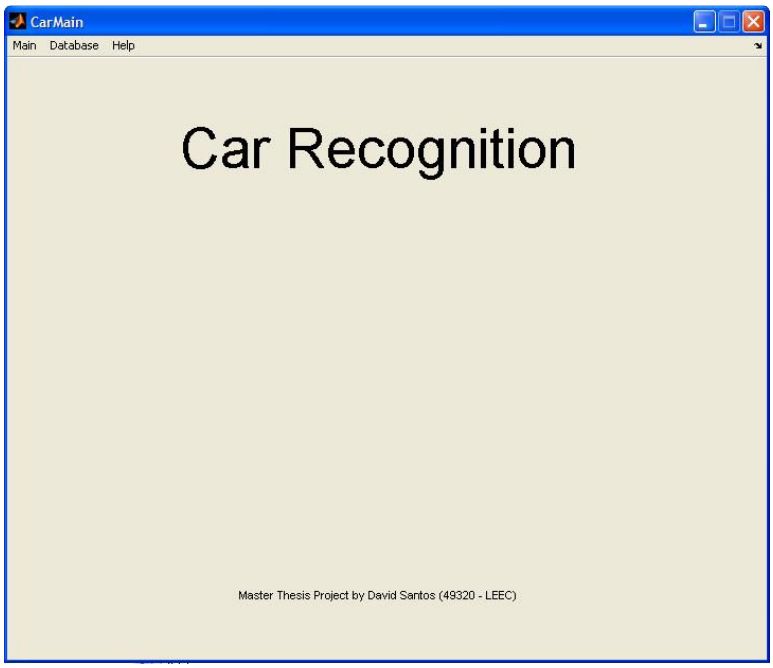


Figure 4.1 - Graphical user interface – initial window.

To start the execution, the user should select one of the three provided menu tabs:

- *Main* (Car Recognition; Exit);
- *Database* (Add to Database);
- *Help*.

An example is shown in Figure 4.2.

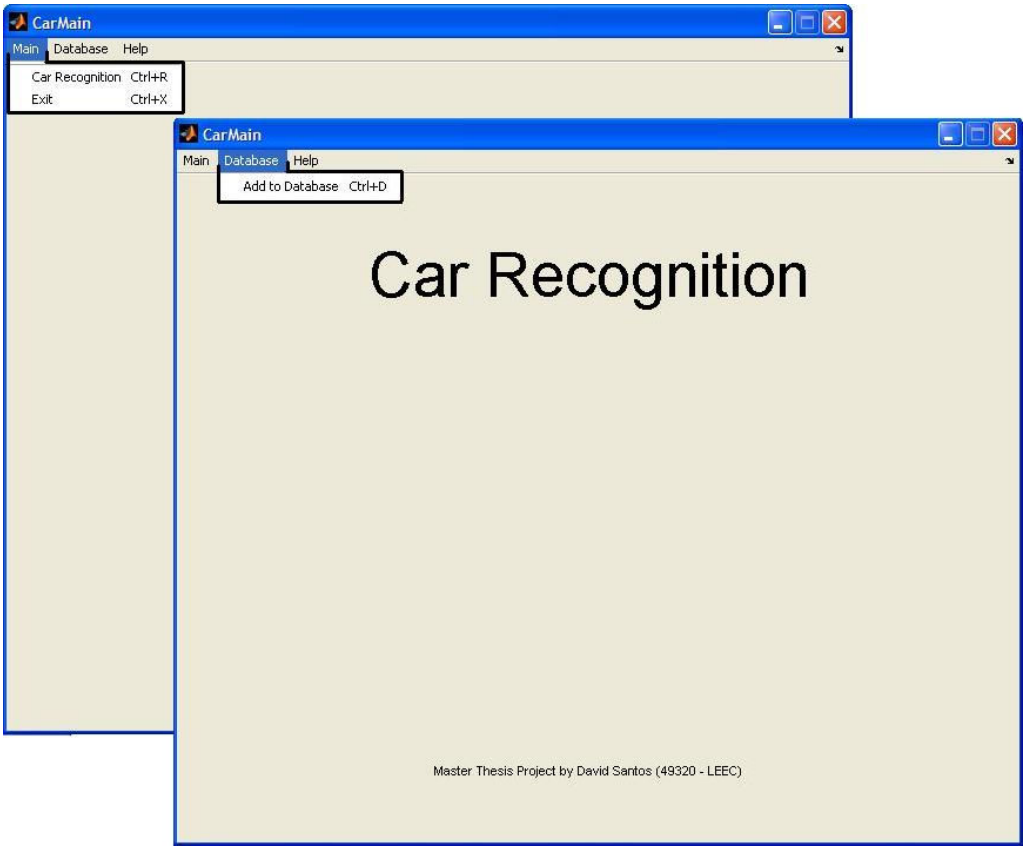


Figure 4.2 - Main and Database menus.

Depending on the selected menu, the Graphical User Interface is adjusted to the specificities of each menu.

4.2.1 *Main* Menu Tab

Here the user is able to select an option called *Car Recognition* or simply terminate the program by selecting the *Exit* option – see Figure 4.2.

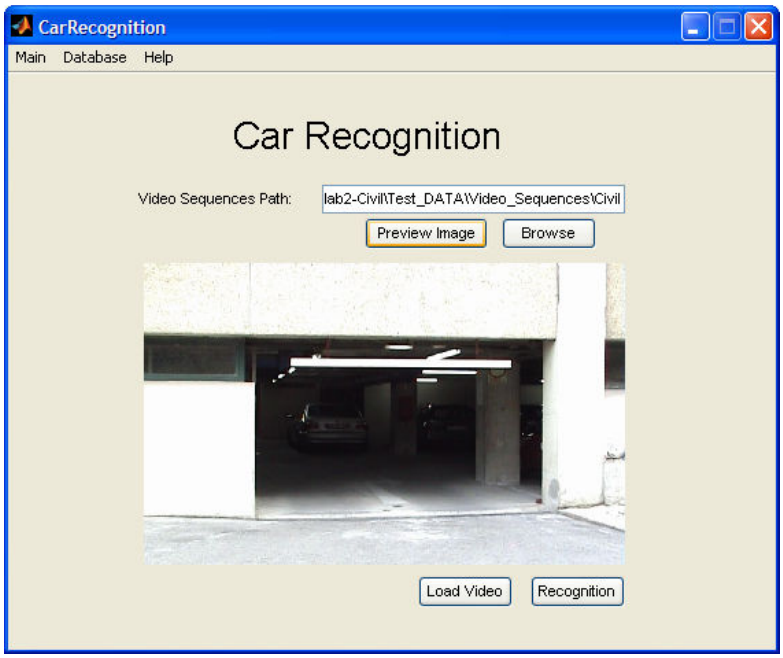


Figure 4.3 - Car Recognition setup window.

The main goal of the program is to recognize vehicles from a loaded video sequence. So, in the *Car Recognition* window, the user chooses which action should be taken, *load* a new *video* sequence, or start the *recognition* of a previously loaded video sequence. Figure 4.3 shows an example of the Car Recognition window.

- **Load Video:** It was decided to use a sequence of individual images as input to the vehicle recognition program, instead of video files. This results in less processing time, since there is no need to convert the video into image files during the execution of the program. The current version of the program only accepts three types of RGB input image file formats: *Portable Networks Graphics* ('.png'), Tagged Image File Format ('TIF') or *Joint Photographic Experts Group* ('.jpg'). The only information required from the user is the directory path of the images sequence to load; the *Browse* button can help the user to locate the input images sequence directory. A preview of an image of the selected sequence can be seen by clicking the *Preview Image* button. The user should then press the *Load Video* button.
  Once the user decides to load a new video the previous one, if loaded, becomes unavailable to the program. It is important to refer that this video could have several vehicles in it, however they should come into camera view one at a time.

- **Recognition:** Once the *Recognition* button is clicked, the vehicle recognition process starts. After each "unknown" vehicle of the loaded sequence is recognized, a window of results will pop-up, showing the three best matching database vehicles and their respective images, names, and similarity distance measures.
  The color of the "unknown" vehicle is also shown, as illustrated in Figure 4.4.

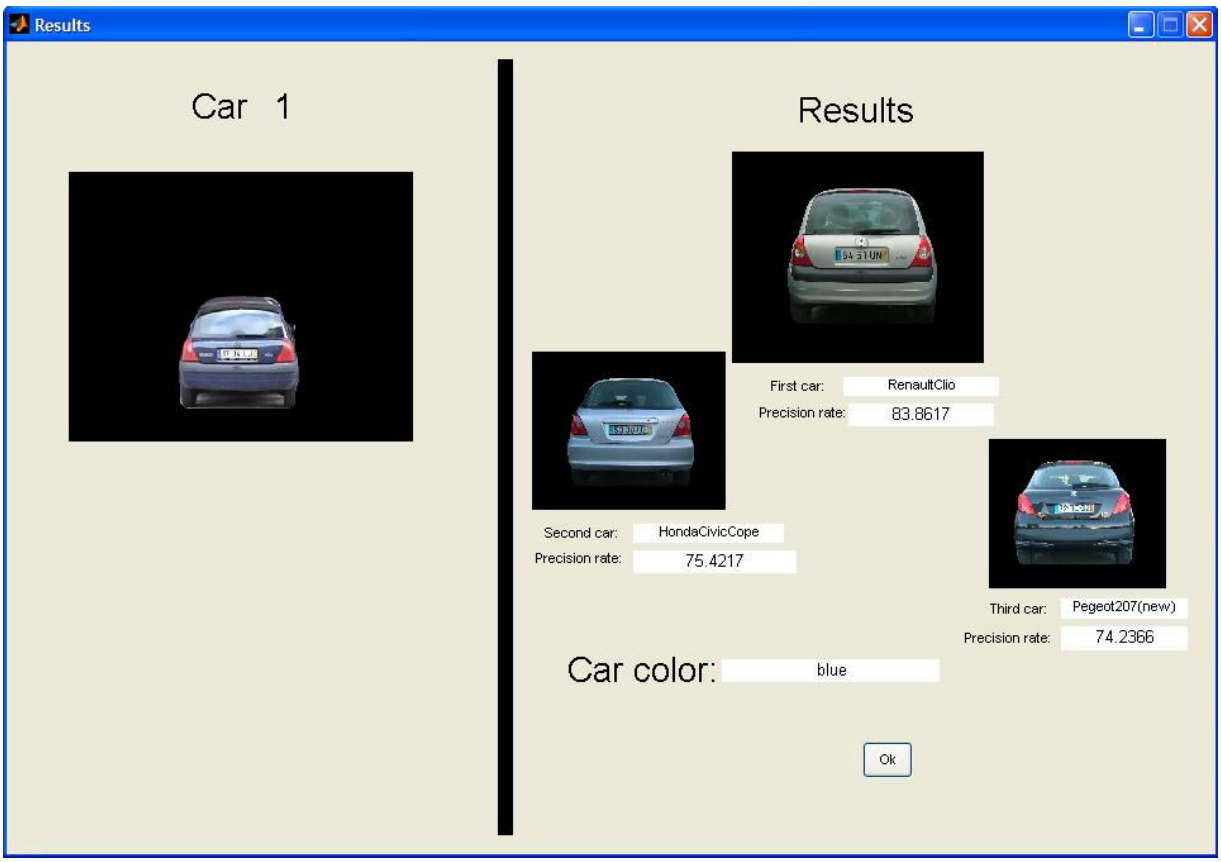


Figure 4.4 - Vehicle recognition results window.

4.2.2 *Database* Menu Tab

When the user selects the "*Add to Database"* option in the *Database* menu tab, he will be allowed to add new vehicle(s) image(s) to the system database. These images should comply with a number of guidelines, as specified below:

- ***Name*:** the name of each file should correspond to the respective manufacturer and model of the vehicle in the image (e.g., "RenaultClio.TIF");

- ***File extension***: The current version of the program only accepts three types of RGB input image file formats: *Portable Networks Graphics* ('.png'), Tagged Image File Format ('TIF') or *Joint Photographic Experts Group* ('.jpg');

- ***Segmented image***: Any image added to the database should be previously segmented and could have an arbitrary size, as illustrated in Figure 4.5.

Figure 4.5 - Example of a database vehicle image.

- ***Vehicle main Color***: In order to improve the performance of the proposed system, the color of all the vehicles added to the database, should not be red. Section 3.2.2 explains the constraints introduced by this color.

After having prepared compliant image files to be added to the database, the user needs to specify the images directory path. Once again, whenever a path is required, a button named *Browse* can help the user to perform the corresponding selection. A preview of an image of the specified folder can be shown by clicking the *Preview Sequence image* button. Figure 4.6 includes an example of an "*Add to Database"* window.

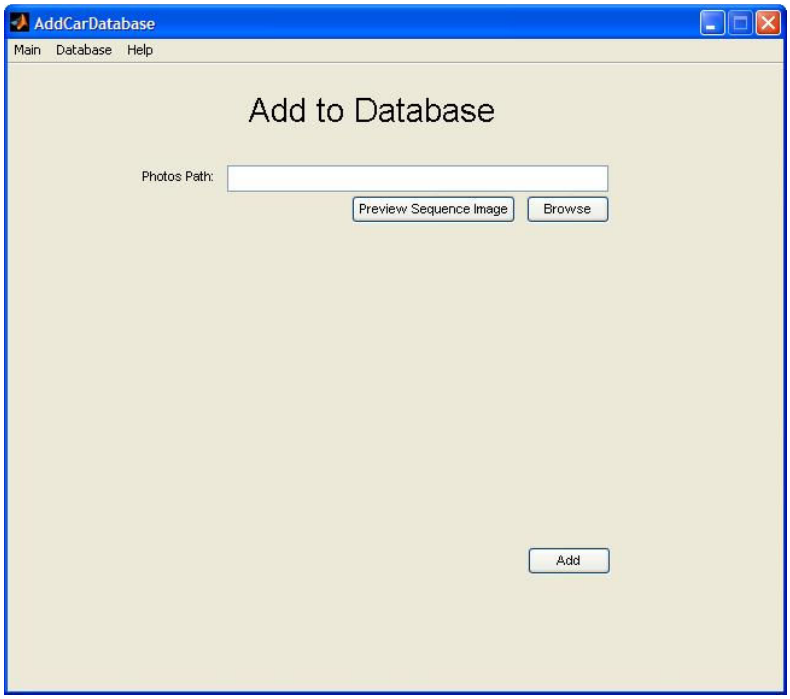


Figure 4.6 - Add to database window.

This database was programmed in order to be easily updated, so whenever the user adds new vehicle images to the database, they will be automatically added to the previous database.

4.2.3 *Help* Menu Tab

Every window has a help menu that the user can access whenever some kind of clarification is needed about that specific operation.

Chapter 5

Results

Based on the widely accepted FERET methodology for the evaluation of recognition algorithms [38], the performance of the vehicle recognition system proposed in this Thesis is listed in terms of the probability *p(k)* that a test sequence (input video sequence) is among the top *k* matches, i.e. the correct recognition rate is the probability at the rank 1 among all the matches in the complete database set. The performance statistics are reported as the cumulative match scores. The rank *k* is plotted along the horizontal axis, and the vertical axis is the percentage of correct matches [38]. The performance of the system can also be confirmed with several tables built for the issue in which it is shown the Rank 1 matches, and respective extracted color of the "unknown" vehicle.

Results Chapter is divided in four sections, in order to conclude about the relevance of the extracted features:

1. *Vehicle Database:* To evaluate the performance of the implemented system, a database was created – see section 5.1 for a detailed description.
2. *Shape Similarity Results*: Results based only on Shape Similarity values;
3. *Lights Similarity Results:* Results based only on Lights Similarity values;
4. *System Results:* Final output of the proposed system, according to equation (12) – see Chapter 3, section 3.4.3.

5.1 IST - Vehicle Database

This database is composed of a test set, a video with 18 vehicles entering in IST vehicle parking garage, and a database set of 40 images of segmented rear views of vehicles from different manufacturers of different models, both created during June 2008 – see Annex 2 - IST Database - Table A.2.1. Although the recording sessions were made on several days, the illumination conditions were very similar. It is Important to refer that test and database sets were acquired without any special requirements in terms of illumination, reflexions or "foreign" shadows reduction.

Some sample images of this database can be seen in Figure 5.1.

Figure 5.1 - Some examples of IST - vehicle database.

5.2 Shape Similarity Results

In Chapter 3, Section 3.2.1, a detailed explanation of the *Vehicle Shape* features extraction was presented; in summary, several features are extracted from the *Vehicle Shape* and then according to equation (10) they are combined - see section 3.4.1.

Figure 5.2 summarizes the system performance based on *Vehicle Shape Similarity*.

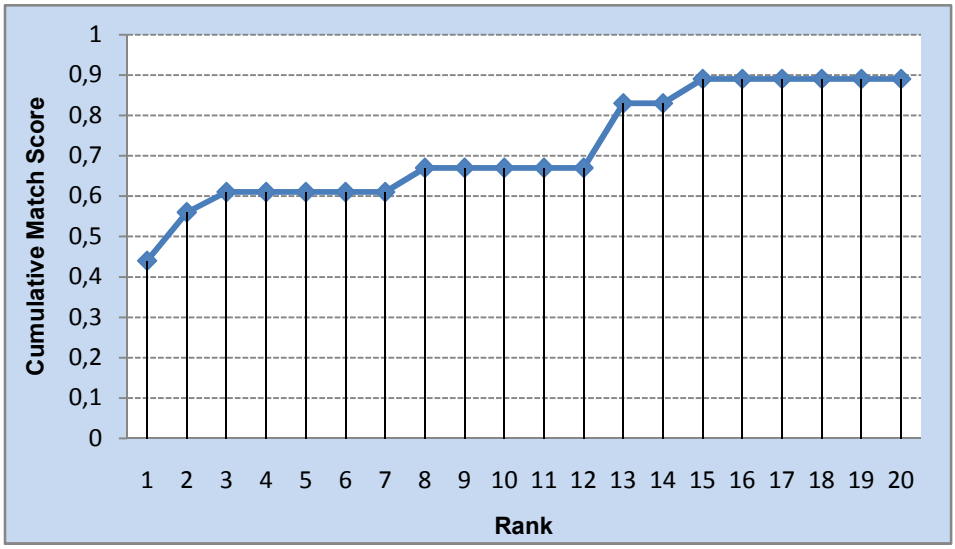


Figure 5.2 - Performance evaluation based on extracted shape features.

The cumulative match score at the top most rank is 44 percent.

Although it is believed that the recognition potential of these features is very high, the quality of the vehicle segmentation has to be very good in order to avoid major errors from the binary segmentation correlation or from inconsistent construction of the distance signals; keep in mind that these signals are relative to the Euclidean distance between the coordinates of the segmented vehicle's boundary pixels (i.e. 360 points corresponding to 360 angles of 1º) and its centroid.

On the other hand, the potential of edge map feature can also be reduced if several shadows and foreign objects appear between the camera and the "unknown" vehicle. Chapter 3, section 3.2.1 presented some improvements done in order to minimize the effect of such adversities.

The following table presents detailed results of the experiments made, indicating, for each "unknown" vehicle the identified manufacturer and model, and the corresponding detected vehicle color. Also the rank of the correct recognition result is included.

Table 5.1 - System results based on extracted shape features.

Video Sequence		System Output		
"Unknown" Vehicle Manufacturer and model	Vehicle Color	Identified Manufacturer and model	Identified Vehicle Color	Rank
Renault Clio	blue	Renault Clio	blue	1
Skoda Fabia	black	Honda Jazz	black or dark gray	2
Peugeot 205	gray	Ford Focus	gray	3
Opel Zafira	blue	Opel Zafira	blue	1
Peugeot 308	gray	Peugeot 308	gray	1
Opel Corsa	white	Opel Corsa	white or light gray	1
Fiat Idea	blue gray	Fiat Idea	blue	1
Honda Accord	blue	Honda Accord	blue	1
VW Sharan	white	VW Sharan	white or light gray	1
Honda Jazz	gray	Honda Jazz	gray	1
Honda HR-V	gray	Audi A3	gray	37
Fiat Punto	gray	Audi A3	gray	8
Mazda 6	dark gray	Skoda Octavia	black or dark gray	13
Peugeot 207 SW	light gray	Rover 400	white or light gray	15
Peugeot 207	dark gray	Fiat Punto	black or dark gray	13
Opel Corsa (old)	red	Citroen C3	red	2
Seat Leon	black	Opel Astra	black or dark gray	13
VW Polo	black	Rover 400	black or dark gray	30

Concluding, it is believed that the results would be much better if the vehicle segmentation was more accurate or the filming area had been specifically prepared to the video shooting, avoiding for example shadows, reflections of the sun in vehicle, foreign objects, vehicle's occluded parts, etc.

Based on Table 5.1 is understandable that *Vehicle Color Extraction* method is well implemented, since all Identified Vehicle Colors are correct.

The cause of the major limitations of these results is illustrated in Figure 5.3, which represents the two cases of poorly segmented vehicles. As can be easily seen, both vehicles present imperfect segmentations, the left one with some missing parts, while the second one, mainly because of some shadows, has foreign objects considered as part of the vehicle.

Honda HR-V VW Polo

Figure 5.3 - Examples of segmented vehicles.

Some of these results are presented with more with detail in Annex 1 - Tables of Experiments.

Table A.1.1, includes images of the “unknown” vehicles and the correspondent top three candidate matching vehicles from the database.

5.3 Lights Similarity Results

The vehicle recognition results obtained using only the features computed form the vehicle back lights and geometrical considerations, using the *Vehicle Lights Similarity* metrics presented, are plotted in Figure 5.4.

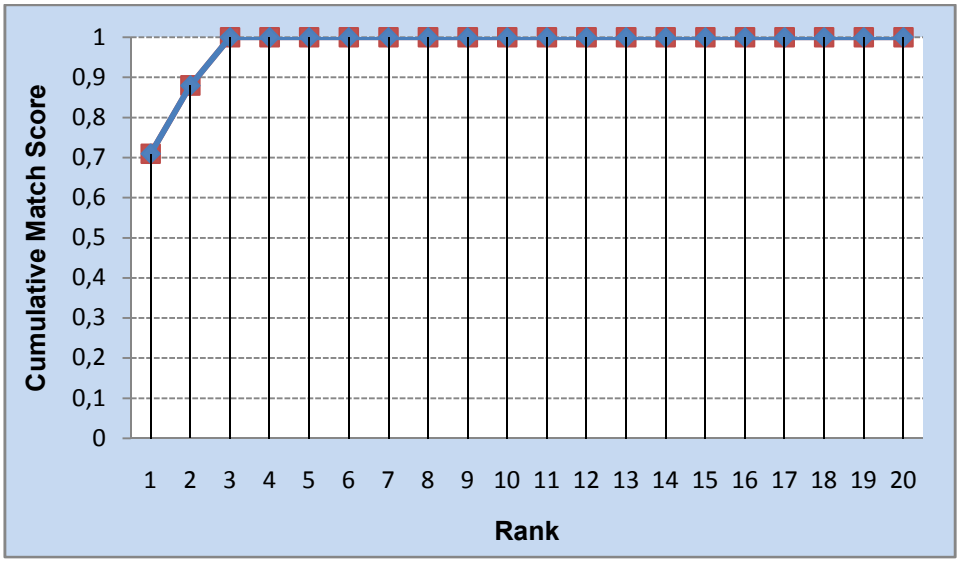


Figure 5.4 - Performance evaluation based on extracted lights features.

Based on extracted lights features the system achieved a correct recognition rate of 71% (12 out of 17 vehicles); it means that the similarity between the light features from the probe and the correct database vehicle are quite evident except for some minor flaws. Confirming that, these features show better results than shape features.

Looking at Table 5.2 the success of the results is very clear, however some errors still occur, mainly because of the similarity between the lights of some vehicles, as for example Peugeot 308 and Peugeot 207 – see Figure 5.5.

Table 5.2 - System results based on extracted lights features.

Video Sequence		System Output		
"Unknown" Vehicle Manufacturer and model	Vehicle Color	Identified Manufacturer and model	Identified Vehicle Color	Rank
Renault Clio	blue	Renault Clio	blue	1
Skoda Fabia	black	Skoda Fabia	black or dark gray	1
Peugeot 205	gray	Audi A3	gray	2
Opel Zafira	blue	Opel Zafira	blue	1
Peugeot 308	gray	Peugeot 207	gray	2
Opel Corsa	white	Fiat Punto	white or light gray	2
Fiat Idea	blue	Opel Corsa	blue	3
Honda Accord	blue	Honda Accord	blue	1
VW Sharan	white	VW Sharan	white or light gray	1
Honda Jazz	gray	Opel Zafira	gray	3
Honda HR-V	gray	Honda HR-V	gray	1
Fiat Punto	gray	Fiat Punto	gray	1
Mazda 6	dark gray	Mazda 6	black or dark gray	1
Peugeot 207 SW	light gray	Peugeot 207 SW	white or light gray	1
Peugeot 207	dark gray	Peugeot 207	black or dark gray	1
Opel Corsa (old)*	red	-	-	-
Seat Leon	black	Seat Leon	black or dark gray	1
VW Polo	black	VW Polo	black or dark gray	1

*lights features were not extracted from this "unknown" vehicle, since its main color is red – see 3.2.2.

Figure 5.5 - Back light similarities – example for Peugeot 308 and Peugeot 207.

An exhaustive representation of these results can be found in Annex 2 - IST Database - Table A.1.2, where some images are shown in order to improve the understanding of the achieved results.

The average elapsed computational time for each "unknown" vehicle recognition is approximately 1 minute and 20 seconds for the first method and 10 seconds for the second (using an Intel® Core (TM) 2 CPU 6400@ 2.13GHz processor with 2Gbyte RAM). These differences are mainly due to the fact that the cross-correlation done between the two *Binary Edge Maps,* in first method, has a high computational cost.

5.4 Vehicle Recognition Results

According to equation (12) – see Chapter 3, section 3.4.3., the overall proposed system performance is estimated, considering both shape and back light features. The performance results are plotted in Figure 5.6.

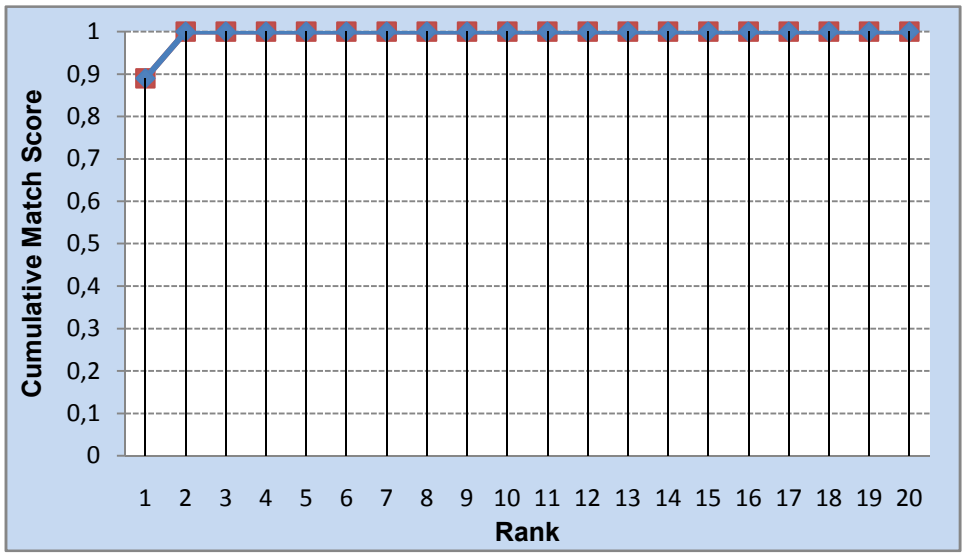


Figure 5.6 - Performance evaluation of proposed system.

The system showed a successful recognition rate of 89% (16 out of 18 vehicles).

By analyzing all results, it is possible to conclude that the *Lights Feature Similarity* method shows better recognition rates than the *Shape Feature Similarity* method for most of the performed experiments. This is mainly due to the fact that the first method requires a good quality vehicle segmentation, while the second copes rather well with this situation, since the vehicle's lights are independently identified. Furthermore, each vehicle has always two symmetrical lights which allow the system to select the best one, avoiding some errors that could arise.

Although the difference between the two methods is clear, it is also possible to notice that their combination improves the performance of the proposed system, over the best individual results – see Table 5.3.

Table 5.3 - System results.

Video Sequence		System Output		
"Unknown" Vehicle Manufacturer and model	Vehicle Color	Identified Manufacturer and model	Identified Vehicle Color	Rank
Renault Clio	blue	Renault Clio	blue	1
Skoda Fabia	black	Skoda Fabia	black or dark gray	1
Peugeot 205	gray	Peugeot 205	gray	1
Opel Zafira	blue	Opel Zafira	blue	1
Peugeot 308	gray	Peugeot 308	gray	1
Opel Corsa	white	Opel Corsa	white or light gray	1
Fiat Idea	blue gray	Opel Corsa	blue	2
Honda Accord	blue	Honda Accord	blue	1
VW Sharan	white	VW Sharan	white or light gray	1
Honda Jazz	gray	Honda Jazz	gray	1
Honda HR-V	gray	Honda HR-V	gray	1
Fiat Punto	gray	Fiat Punto	gray	1
Mazda 6	dark gray	Mazda 6	black or dark gray	1
Peugeot 207 SW	light gray	Peugeot 207 SW	white or light gray	1
Peugeot 207	dark gray	Peugeot 207	black or dark gray	1
Opel Corsa (old)	red	Citroen C3	red	2
Seat Leon	black	Seat Leon	black or dark gray	1
VW Polo	black	VW Polo	black or dark gray	1

In terms of computational cost, approximately 1.5 minutes is required for each vehicle recognition operation.

Like for the previous sections of this chapter, an exhaustive representation of these results can be found in Annex 1 - Tables of Experiments - Table A.1.3.

Looking at the previous tables, Table 5.1 and Table 5.2, there are several examples where wrong results with one of the similarity methods are improved with the combination with the other. As an example we have a Peugeot 308 that failed the lights similarity, but with the features combination achieved a Rank 1 recognition, while a Seat Leon had a wrong recognition based on *shape similarity* and based on the combination method was correctly recognized – see Figure 5.7.

"unknown Vehicle"	Shape similarity Rank1	Lights Similarity Rank1	System Rank1
			
Peugeot 308	Peugeot 308	Peugeot 207	Peugeot 308
			
Seat Leon	Rover 400	Seat Leon	Seat Leon

Figure 5.7 - Example of similarity combination advantages.

Chapter 6

Conclusions

Vehicle external features are particular to each model, allowing to distinguish one model from the others. Many studies in different areas such as safety (crime prevention, surveillance problems), intelligent transport systems (driver assistance systems, intelligent parking systems) and traffic management (traffic parameters), continue to suggest that a vehicle recognition system is a good manner to help solving some of these problems.

Although software implementations of vehicle detection systems have often been reported in the literature, the vehicle recognition issue is still at its infancy, with the usual systems relying exclusively on license plate recognition. However, the most recent approaches show that the evolution of computer vision techniques is increasing the ability to build a robust vehicle recognition system based on its external features, allowing to recognize also the manufacturer and model.

The main advantage over other vehicle recognition systems comes from the fact that vehicle external features can't be easily changed like for example its license plate, since they are components of the vehicle. Even more, the recognition can be done at a distance and with relatively low video quality images.

This project focused on the implementation of two novel vehicle external features methods developed by the author: *Vehicle Shape Features* and *Vehicle Lights Features* that are then combined in order to obtain a robust vehicle recognition system. The first method evidences external shape of the vehicle, especially its measures, edges and shape contour. The second method considers the vehicle lights features by performing several ways to classify them, its orientation, eccentricity, position, angle with vehicle license plate and its shape contour.

A large number of experimental results, with a database specifically created for this project, have demonstrated that both methods show encouraging recognition rates, even though the first method has proven to be less reliable than the second, notably if the video acquisition conditions are not controlled. This is mainly due to the first method's proneness to fail when the quality of vehicle segmentation is not so good, while the second is more robust, even if not totally unaffected by this type of errors. Therefore, better results would be expected if more accurate vehicle segmentation could be achieved.

Nevertheless, the combined results show that when system makes usage of both methods, its performance is clearly increased, from around 45% and 70% of recognition rates for first and second methods respectively, to around 90% for the combined system.

In terms of computational cost, approximately 1.5 minutes were required for each vehicle recognition operation, using an Intel® Core (TM) 2 CPU 6400@ 2.13GHz processor with 2Gbyte RAM. The average elapsed computational time for each "unknown" vehicle recognition is approximately 1 minute and 20 seconds for the first method and 10 seconds for the second.

Although the achieved results are very encouraging, more experiments should be made to validate the proposed system.

In order to do it, several difficulties should be overcome, such as:

- System Database: The system database should be more complete, including more examples per vehicle (manufacturer and model) to accurately extract their features. To facilitate one of the vehicles' database demands, an agreement could be done with vehicles companies. And so instead of using segmented real views of the vehicle, the program should be able to use the drawings offered by vehicle companies. This could be also a manner for the system database stay updated;
- Video Scenario: The filming conditions should be enhanced, for example, by building a structure that could minimize some unwanted segmentation errors, resulting for instanced from reflections, wrong vehicle's position or overlapping vehicles – see Figure 3.2.

In the future, a robust vehicle recognition system could play an important role in global security by fighting crime, and eventually preventing terrorism. This system could be a part of a wide variety of applications that include visual surveillance, and access control, in special environments like motorways or parking lots.

In what concerns the work here developed, there are some improvements that could be included in future work:

- Computational Cost: As presented in previous chapter, the difference between the computation costs of the two implemented method is enormous; the second method is close to real-time while the first one has a cost of over one minute. Based on the fact that this high computational cost is mainly due to the correlation done between the two *Binary Edge Maps*, a faster correlation method should be implemented.
- Improve Video Segmentation: As has been written several times in this work, one of its main flaws is the video segmentation quality, due to the filming conditions and also to the fact that the implemented method is quite simple. Thus, one future proposal for this work is its combination with a more robust video segmentation method in order to increase some of the achieved results.

Yet, and when a precise vehicle segmentation is performed, it is believed that the proposed system can be used as a reliable standalone control system, ensuring high recognition performance.

Concluding, it was proven that the novel and original methods developed by the author can be exploited as feasible techniques for automatic vehicle recognition and have an imperative contribution to the "*work in progress*" vehicle recognition approach.

Annex 1

Tables of Experiments

The following tables present detailed results of the experiments made, showing several images in order to best understand the recognition system performance. Table A.1.1 is related to results based on extracted vehicle shape features, while results based on the extracted lights features are shown in Table A.1.2. Finally, it presents detailed results of the proposed recognition system in Table A.1.3.

Table A.1.1 - Recognition results based on extracted shape features.

"unknown" segmented vehicle	First Identified database vehicle	Second Identified database vehicle	Third Identified database vehicle
			
Renault Clio	Renault Clio	Peugeot 307	Fiat Punto
			
Skoda Fabia	Honda Jazz	Skoda Fabia	Ford Focus
			
Peugeot 205	Ford Focus	Mistubishi Colt	Peugeot 205
			
Opel Zafira	Opel Zafira	Ford Focus	Skoda Fabia
			

Peugeot 308	Peugeot 308	Renault Clio	Opel Astra
			
Opel Corsa	Opel Corsa	Opel Astra	VW Polo
			
Fiat Idea	Fiat Idea	Ford Focus	Mistubishi Colt
			
VW Sharan	VW Sharan	Honda HR-V	Opel Zafira
			
Peugeot 207 SW	Rover 400	Honda Civic Cope	Peugeot 207
			
Seat Leon	Rover 400	Honda Civic Hybrid	BMW 320

Table A.1.2 - Recognition results based on extracted lights features.

“unknown” segmented vehicle	First Identified database vehicle	Second Identified database vehicle	Third Identified database vehicle
			
Renault Clio	Renault Clio	Skoda Fabia	Honda Civic Cope
			
Skoda Fabia	Skoda Fabia	Skoda Octavia	Renault Clio
			
Peugeot 205	**Audi A3**	Peugeot 205	BMW 320
			
Opel Zafira	Opel Zafira	Skoda Octavia	BMW 316
			
Peugeot 308	Peugeot 207	Peugeot 308	Honda Civic Cope

			
Opel Corsa	Fiat Punto	Opel Corsa	Mistubishi Colt
			
Fiat Idea	Opel Corsa	Opel Astra	Fiat Idea
			
VW Sharan	VW Sharan	Honda Accord	Honda Civic Hybrid
			
Peugeot 207 SW	Peugeot 207 SW	VW Polo	Toyota Yaris
			
Seat Leon	Seat Leon	Audi A3	Honda Civic Hybrid

Table A.1.3 - Overall recognition results.

“unknown” segmented vehicle	First Identified database vehicle	Second Identified database vehicle	Third Identified database vehicle
			
Renault Clio	Renault Clio	Honda Civic Cope	Skoda Fabia
			
Skoda Fabia	Skoda Fabia	Skoda Octavia	Renault Clio
			
Peugeot 205	Peugeot 205	Audi A3	BMW 320
			
Opel Zafira	Opel Zafira	Fiat Idea	Skoda Octavia
			
Peugeot 308	Peugeot 308	Peugeot 207	Honda Civic Cope

			
Opel Corsa	Opel Corsa	Fiat Punto	Mistubishi Colt
			
Fiat Idea	Opel Corsa	Fiat Idea	Opel Astra
			
VW Sharan	VW Sharan	Honda Accord	Audi A3
			
Peugeot 207 SW	Peugeot 207 SW	VW Polo	Rover 400
			
Seat Leon	Seat Leon	Audi A3	Honda Civic Hybrid

Annex 2

IST Database

The following table presents details of the IST database created for this Thesis.

Table A.2.1 - IST database set – list of vehicles.

 Audi A3	 Mazda 3	 Peugeot 206
 Audi A3 (old)	 Mazda 6	 Peugeot 307
 BMW 316	 Mitsubishi Colt	 Renault Clio
 BMW 320	 Opel Astra	 Rover 25
 Citroen C3	 Opel Astra (old)	 Rover 400

Fiat IDEA

Opel Corsa

Seat Leon

Fiat Punto

Opel Corsa (old)

Skoda Fabia

Ford Focus

Opel Frontera

Skoda Octavia

Ford S-max

Opel Zafire

Toyota Corolla

Honda Civic Cope

Peugeot 207

Toyota Yaris

Honda Civic Hybrid

Peugeot 207 SW

VW Golf 1.9

Honda Jazz

Peugeot 308

VW Polo tdi

Honda HR-V

Peugeot 205

VW Sharan

Honda Accord

References

[1] Z. Sun, G. Bebis, and R. Miller, *"On-Road Vehicle Detection: A Review"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20-21 May 2006, pp. 694-711, vol.28.

[2] T. Kalinke, C. Tzomakas, and W. von Seelen, *"A texture-Based Object Detection and an Adaptative Model-based Classification",* Proceedings IEEE International Conference Intelligent Vehicles, 1998, pp. 143-148.

[3] J. Crisman and C. Thorpe, *"Color Vision for Road Following"*, Proceedings of the SPIE Conference Mobile Robots, November 1988, pp. 246-249.

[4] A. Gianchetti, M. Campani, and V. Torre, "*The Use of Optical Flow for Road Navigation*"**,** IEEE Trans. Robotics and Automation, no. 1, pp. 34-48, vol.14.

[5] W. Kruder, W. Enkelmann, and S. Rossle, "*Real-time Estimation and Tracking of Optical Flow Vectors for Obstacle Detection"* Proceedings IEEE Intelligent Vehicles Symp., 1995, pp. 304-309.

[6] U. Franke, and I. Kutzbach, "*Fast Stereo Based Object Detection for Stop and Go Traffic",* Intelligent Vehicles, 1996, pp. 339-344.

[7] L. Jin, B. Gu, R. Wang, L. Guo, Y. Zhao and L. Li, "*Preceding Vehicle Detection Based on Multi-characteristics Fusion",* IEEE, 2006, pp. 356-360.

[8] X. Wen, H. Zhao, N. Wang and H. Yuan, "*A Rear-Vehicle Detection System for Static Images Based on Monocular Vision",* IEEE, ICARCV 2006.

[9] C. Tzomakas, and W. von Seelen, "*Vehicle Detection in Traffic Scenes Using Shadows",* Technical Report 98-06, Institut fur Neuroinformatik, Ruht-Universitat, Bochum, Germany, 1998.

[10] S. Shi, Z. Qin, and J. Xu, "*Robust Algorithm of Vehicle Classification",* International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, pp. 269-272.

[11] Y. Du and N. Papanikolopoulos, "*Real-Time Vehicle Following Through a Novel Symmetry-Based Approach",* IEEE International Conference on Robotic & Automation, 1997, pp. 3160-3165, vol.4.

[12] H. Chang, C. Fu, and C. Huang, "*Real-Time Vision-Based Preceding Vehicle Tracking and recognition",* IEEE Intelligent Vehicles Symposium, 2005, pp. 514-519.

[13] A. Kuehnle, "*Symmetry-Based Recognition of Vehicles Rears",* Pattern Recognition Letters, 1991, pp. 249-258 vol.12, no, 4.

[14] A. Santhanam and M. M. Rahman, "*Moving Vehicle Classification Using Eigenspace",* Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, 9-15 Oct. 2006, pp. 3849-3854.

[15] R. C. Gonzalez, Paul, and Wintz, "*Digital Image Processing*", Addison-Wesley Publishing Company Limited, 1986.

[16] X. Ma, and W. Grimson, "*Edge-based rich representation for vehicle classification",* Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05), 2005.

[17] D. G. Lowe, "*Distinctive image features from scale-invariant keypoints",* Accepted for publication in the International Journal of Computer Vision, Jan. 2004.

[18] J. Migdal and W. E. L. Grimson, "*Background subtraction using Markov thresholds",* IEEE Workshop on Motion and Video Computing, Jan. 2004.

[19] C. Stauffer and W. E. L. Grimson, "*Adaptive background mixture models for real-time tracking",* Proc. CVPR, 2:246-252, 1999.

[20] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. Seelen, "*An Image Processing System for Driver Assistance",* Image and Vision Computing, 2000, vol.18 no.5.

[21] M. Bertozzi, A. Broggi, and S. Castelluccio, "*A Real-Time Oriented System for Vehicle Detection",* J. Systems Architecture, 1997, pp. 217-325.

[22] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miche, S. Mousset, and G. Moulminet, "*A Cooperative Approach to Vision-Based Vehicle Detection",* IEEE intelligent Transportation Systems, 2001, pp. 209-214.

[23] Sen-Ching S. Cheung and Chandrika Kamath, "*Robust Techniques for Background Subtraction in Urban Traffic Video",* Visual Communications and Image Processing 2004. Proceedings of the SPIE, Volume 5308, pp. 881-892.

[24] L. Dlagnekov, "*Video-Based Car Surveillance: License Plate Make and Model Recognition",* Master's Thesis, University of California at San Diego, 2005, taken from *www-cse.ucsd.edu*.

[25] David Anthony Torres, "*More Local Structure Information for Make-Model Recognition",* Dept. of Computer Science, University of California at San Diego, taken from http://www-cse.ucsd.edu.

[26] Y. Gao and M. K. H. Leung, "*Face Recognition using Line Edge Map",* In IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, *pp.764-779 Vol. 24*.

[27] Milos Stojmenovic, "*Real Time Machine Learning Based Car Detection in Images with Fast Training",* Springer-Verlag, March 2006.

[28] Viola, P., Jones, M., "*Robust real-time face detection",* Int. J. Computer Vision 57(2), 2004, 137–154.

[29] D. Lowe, "*Distinctive image features from scale-invariant keypoints", IJCV*, 2(60):91–110, 2004.

[30] N. Otsu "*A Threshold Selection Method from Graylevel Histograms*", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, Jan. 1979, pp. 62--66.

[31] D. Santos, and J. Fazenda, "*Human Detection and Recognition using Video Information",* Technical University of Lisbon, Portugal, 2005.

[32] En.wikipedia: *http://en.wikipedia.org/wiki/HSL_and_HSV*

[33] L. Wang, T. Tan, H. Ning, and W. Hu, *"Silhouette analysis-based gait recognition for human identification",* IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 25 No. 12, December 2003, pp. 1505-1518.

[34] C. Paulo, "*Detection and Recognition of Traffic Signs*"**,** Master Thesis, IST, Technical University of Lisbon, Portugal, 2007.

[35] En.wikipedia: *http://en.wikipedia.org/wiki/Euclidean_distance*

[36] En.wikipedia: *http://en.wikipedia.org/wiki/Mahalanobis_distance*

[37] J. Lewis, *"Fast Normalized Cross-Correlation",* Industrial Light & Magic, 1995, taken from http://citeseerx.ist.psu.edu.

[38] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "*The FERET Evaluation Methodology for Face-Recognition Algorithms*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, October 2000, pp. 1090-1104.